
LOAD
REVISTA PARA USUARIOS DE
MSX
AÑO 1 Nº 2 ₳ 2.00 REP. ARGENTINA
EL SISTEMA OPERATIVO DE LA DISKE
SOFTWARE:
COMERCIAL
ENTRETENIMIENTO
EDUCATIVO
PROGRAMACION APLICADA
EL MODEM TMX-510
PODEROSOS ARCHIVOS
MSX DPC-200

¡YA!
CLUB DE USUARIOS
Talent MSX
Cabildo 2027 - 1° piso
Capital

dialogo - D. P.



El Club de Usuarios de MSX

Ya funciona en su nueva dirección: Cabildo 2027 - 1° piso Capital

Invitamos a los felices usuarios de la TALENT MSX al **curso gratuito** de introducción al fabuloso mundo de MSX.

Participe del Club de Usuarios de MSX y encuéntrese con sus amigos que también tienen la TALENT MSX, e intercambiará programas, datos y chimentos.

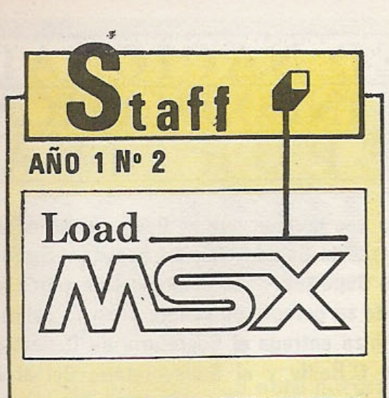
Podrá probar todos los accesorios de la línea MSX, ¡¡desde disketteras hasta robots!!

Podrá ver y leer todo lo que le interese sobre la norma MSX: catálogos, libros y revistas de todo el mundo. Todo con la seguridad, respaldo y seriedad que sólo TALENT puede brindarle.

¡Para inscribirse, no olvide traer su factura de compra!

Club Talent MSX

MSX es marca registrada de MICROSOFT CORPORATION.



Director General
Ernesto del Castillo

Director Editorial
Cristian Pusso

Director Periodístico
Fernando Flores

Director Financiero
Javier Campos Malbrán

Arte y Diagramación
Fernando Amengual
Tamara Migelson

Coordinador
Ariel Testori

Redacción
Eduardo Mombello
Andrea Sabin Paz

Departamento de Avisos
Oscar Devoto

Departamento de Publicidad
Guillermo González Aldalur

Departamento fotográfico
Víctor Grubicy

Load Revista para usuarios de MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5º Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Registro Nacional de la Propiedad Intelectual: E.T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual.

Todos los derechos reservados.

Precio de este ejemplar: A 2
Impresión: Calcotam. Fotocromo tapa: Columbia. Fotocomposición: Van Waveren.
Los ejemplares atrasados se venderán al precio del último número en circulación.

Prohibida la reproducción total o parcial de los materiales publicados, por cualquier medio de reproducción gráfico, auditivo o mecánico, sin autorización expresa de los editores. Las menciones de modelo, marcas y especificaciones se realizan con fines informativos y técnicos, sin cargo alguno para para las empresas que los comercializan y/o los representan. Al ser informativa su misión, la revista no se responsabiliza por cualquier problema que pueda plantear la fabricación, el funcionamiento y/o la aplicación de los sistemas y los dispositivos descriptos. La responsabilidad de los artículos firmados corresponde exclusivamente a sus autores.

Distribuidor en Capital: Martino, Juan de Garay 358, P.B. Capital.
Distribuidor interior: DGP: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800.

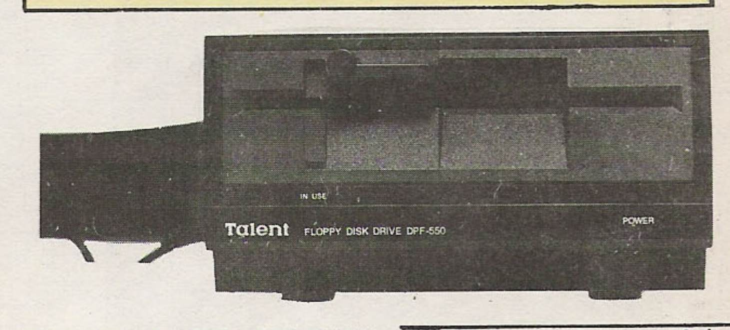
Sumario

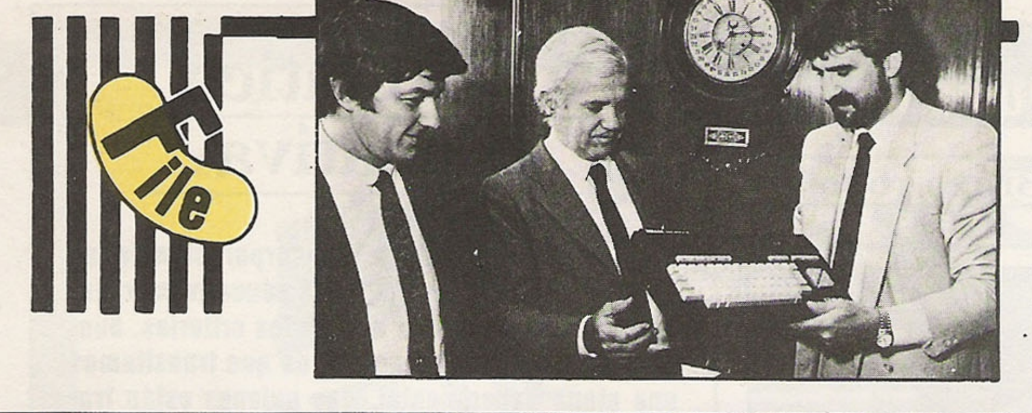
Informática educativa

Hoy muchos se plantean la incorporación de las computadoras en los institutos educativos argentinos. El tema da lugar a distintos criterios, aunque hay coincidencia acerca de que transitamos una etapa experimental. Hay quienes están trabajando seriamente en este campo, como el Centro para el Desarrollo de la Inteligencia, que utiliza los equipos Talent MSX para difundir los múltiples usos del computador. Pero quizá lo más importante es que capacita y perfecciona al docente, y apunta a que el profesional domine el uso de nuevas herramientas. Porque no se puede tratar esta cuestión solamente en términos de hardware. El equipamiento puede resultar inútil si maestros y profesores no saben cómo aplicar la máquina y los programas.

Por eso resulta tan importante el esfuerzo que está realizando la empresa Telemática para que la introducción de las tecnologías sea acompañada por una inteligente formación de los educadores.

LOS EDITORES





RED NACIONAL DE INFORMATICA DEPORTIVA

Durante una reunión que se llevó a cabo en el Ministerio de Salud y Acción Social —Secretaría de Deportes—, la empresa Telemática, a través de su gerente de ventas Juan C. Zahrebelnyj, hizo entrega al Secretario de Deportes Rodolfo O'Reilly y al Subsecretario del área Osvaldo Otero, un equipo completo de computación marca TALENT MSX. Lo hizo en calidad de colaboración para el inicio de las tareas de exploración con vistas a una Red Nacional de Informática Deportiva que en un futuro próximo abarcará todo el territorio de la República.

El equipo está compuesto por una computadora, una pantalla (T.V. color 14''), un drive de disquetes DPF 550; una caja de disquetes de 5 1/4'' y un modem para teléfono.

La Red Informática Deportiva será instrumentada por la Secretaría de Deportes a través de la Dirección Nacional de Promoción del Deporte y la Recreación. El área de Deporte no sólo está interesada en la modernización de los sistemas de archivos y detección de recursos de infraestructura, físicos y humanos relacionados con el deporte sino que tratará a través de la Red Nacional de brindar a todo el país un modelo de sistema educativo-deportivo para niños y adolescentes. Los equipos terminales ubicados en todas las capitales provinciales permitirán transmitir y recibir información vital para este propósito.

EL KEY-PAD DE TALENT

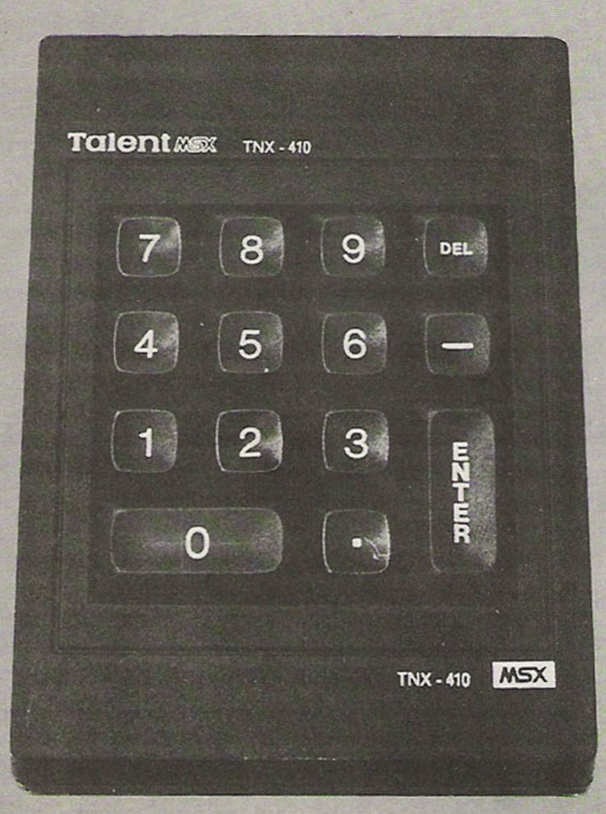
Nuevamente TELEMATICA sorprende con un nuevo desarrollo para la Talent MSX.

Se trata ahora de un teclado numérico con los diez dígitos decimales y una tecla de ENTER. Está diseñado para ser conectado como un joystick a uno de los pórticos que la Talent MSX posee para ese fin.

Pero el Key-pad no viene solo, demostrando otra vez la seriedad y capacidad que caracteriza a esta empresa.

Ha desarrollado un soft de apoyo, que almacenado en disco, permite que el teclado sea leído en todo momento. De esta forma nuestra DPC-200 reconocerá (una vez cargado este programa) a los números pulsados en el key-pad de la misma forma que reconoce a los números que incluye el teclado original.

De más está mencionar las utilidades de este aparato, pero seguramente, una vez que lo hayan adquirido, se olvidarán de la primer fila del teclado de la DPC-200.



En el número anterior comentamos una de las novedades de TALENT en materia educativa: la RED LOCAL EDUCACIONAL MINI-LAN.

Ahora incluimos algunas modificaciones.

Estas son:
1) Velocidad de comunicaciones: 30.000 bits/segundo.
2) Costo adicional por estación alumno: A 100.-
3) Costo adicional por estación maestra: A 150.-

Compu Prando
Av. de Mayo 965 - 1084 - Bs. As.
Tel.: 38-0295

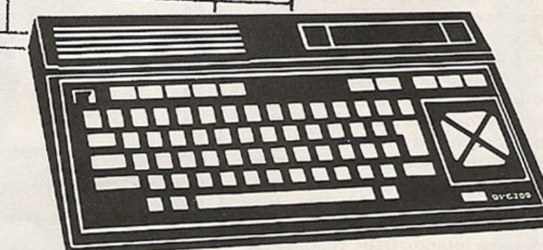
CONOZCA LA DIFERENCIA...

HARDWARE
.EQUIPOS
.PERIFERICOS
.ACCESORIOS

SOFTWARE
.UTILITARIOS
.LENGUAJES
.GESTION
.EDUCACION
.JUEGOS

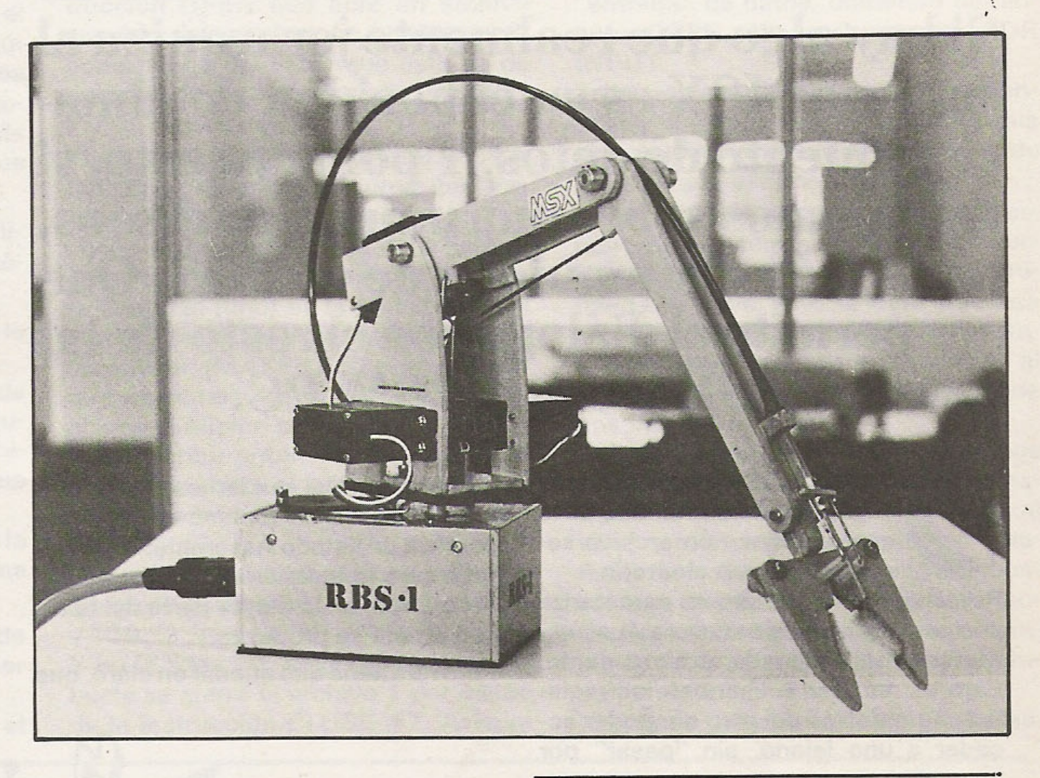
CURSOS
LIBROS





BRAZO ROBOT RBS-1 PARA COMPUTADORAS MSX

La robótica está surgiendo como una de las ideas claves para aplicar las nuevas tecnologías de fabricación de la década presente. Es una tecnología multidisciplinaria que abarca los últimos avances en la tecnología de control por microcomputadoras, programación, física e ingeniería electrónica, mecánica y eléctrica. La integración de estas disciplinas permite la generación de un renacimiento en la tecnología de fabricación.
Las mejoras en la productividad mediante la automatización de las fábricas, manufacturas computarizadas integradas, y la mayor posibilidad de trabajar en medios hostiles son sólo la "punta del iceberg".



Presentamos el brazo robot MSX RBS-I

El sistema RBS-I permite acceder a un bajo costo a un sistema robótico muy útil para ingenieros, maestros, estudiantes técnicos para aprender y experimentar con robots.
Se puede manejar con cualquier computador MSX y ayuda a convertirse en uno de los innovadores del mañana en robótica, automatización y control de movimiento.
Producido por BaireSoft y Prosoft, el diseño, la construcción y el software: estuvo a cargo de Emilio D. Graffigna, Roberto N. Tokuda y, Hugo D. Caro

En educación

Dado que RBS-I se conecta directamente a impresora, las amplias facilidades de MSX permiten controlarlo con todos los lenguajes disponibles para estas máquinas, como ser MSX-BASIC, MSX-LOGO, FORTRAN, LISP, FORTH, etc. Une a la experimentación "mecánica" la introducción al mundo de la inteligencia artificial, con programas que pueden resolver problemas de tiempo y movimiento con "Sentido Común".
Para profesores que deseen desarrollar un currículum sobre reparación y operación de robots, RBS-I es una excelente herramienta.
Debido a su bajo costo, RBS-I puede hacer sentir su presencia y enriquecer las clases de computación de escuelas primarias, secundarias y universidades.

En la industria

Los ingenieros que deseen diseñar sistemas automatizados encontrarán en el RBS-I una herramienta de prueba y verificación muy útil.
La verificación y armado de sistemas multi-brazo, la llamada "coreografía" de producción, y las consideraciones de programación se pueden probar, contruír modelos, modificar y verificar con RBS-I, a un costo realmente bajo en comparación con sistemas de mayor envergadura.

Descripción del producto

El sistema completo RBS-I para MSX consta de:

- El brazo robot, construído en materiales resistentes, de gran duración y versatilidad y totalmente autocontenido.
- Cable de conexión, que se coloca directamente a la salida de impresora.
- Un cassette de programa, que permite la programación desde MSX-BASIC de secuencias y posiciones.
- Un manual teórico y de uso.

Especificaciones

Configuración: 4 grados de libertad, movimientos: brazo, antebrazo, giro torre y abrir/cerrar pinzas.

Motores:
4 servomotores Futaba S128, 3 Kg/cm.

Rango de movimientos:
Torre: +/− 180 grados
Antebrazo: +/− 45 grados
Brazo: +/− 90 grados
Pinza: 0 a 4,5 cm.
Capacidad de carga: 250/300 gramos
Velocidad: 0.24 seg/60 grados

Electrónica general:
Fuente: 6 V regulada/transformador 220 V/50 Hz
Rango de temperaturas: 0 a 45 ºC

Dimensiones:
Altura máxima: 35 cm
Ancho: 7 cm
Largo base: 11 cm

El RBS-I—distribuido por Prosoft— estará próximamente a disposición del público quien podrá adquirirlo directamente en la red de Telemática, fabricantes de la computadora Talent MSX. Cabe destacar que es un desarrollo íntegramente argentino, con el mínimo material importado, empleando los de mejor calidad para ser coherentes con la filosofía del producto: "bajo costo y elevado standard de calidad".

TODO EL PODER DE LOS ARCHIVOS

Si hay algo que realmente jerarquiza al sistema MSX es su capacidad de almacenamiento de datos. Y por supuesto, si trabajamos sobre una unidad de discos, nada más distinguido que el DOS. Que nos permite el diálogo con el resto de las marcas MSX y con la IBM PC.



Seguramente aunque sea muy vaga, tenemos una idea de lo que significan un archivo secuencial y uno aleatorio.

Repasemos. El primero se caracteriza porque la forma de acceso a él es estrictamente ordenada y ascendente. Esto es: se leen o escriben los registros, uno detrás del otro sin poder acceder a uno lejano, sin "pasar" por los anteriores.

Esta es la idea básica que rige a este tipo de archivos.

Los llamados archivos aleatorios se distinguen por permitir el acceso a cualquiera de sus registros, simplemente invocando su número.

Así podremos acceder a cualquiera de los registros de un archivo sin tener que leer los anteriores.

Cuando los que saben crean un archivo, lo hacen con alguna de estas dos características. Es decir que si graban uno en forma secuencial es porque así han de leerlo luego, lo que no quiere decir que su archivo no puede ser leído o tratado como uno aleatorio.

Pongamos algo de luz en este asunto. Un archivo es un archivo, ¿está claro? Esto quiere decir que cualquier cosa que se encuentre en el directorio de nuestro diskette es un conjunto de números y puede ser leído como tal, sin importar si es un programa BASIC, Assembler, el mismísimo MSX-DOS o simplemente datos.

De esta forma podremos tomar tanto como un archivo secuencial o como uno aleatorio a cualquier paquete de números que esté en el disco y tenga, por supuesto, algún nombre.

Con eso nos sobra para considerarlo un archivo.

Lo que cambiará en cada caso será el significado que dicha lista de datos tenga.

Seguramente si queremos interpretar como texto a un archivo que corresponde a un listado Assembler, el resultado será la incoherencia más grande del mundo, en la mayor parte del texto que de allí se produzca.

Pero nos tiene que quedar en claro, que con la diskettera debajo de un brazo y el MSX-DOS debajo del otro, podremos descuartizar cualquier cosa que se encuentre en un disco en cuyo terreno gobierne un DOS más o menos decente.

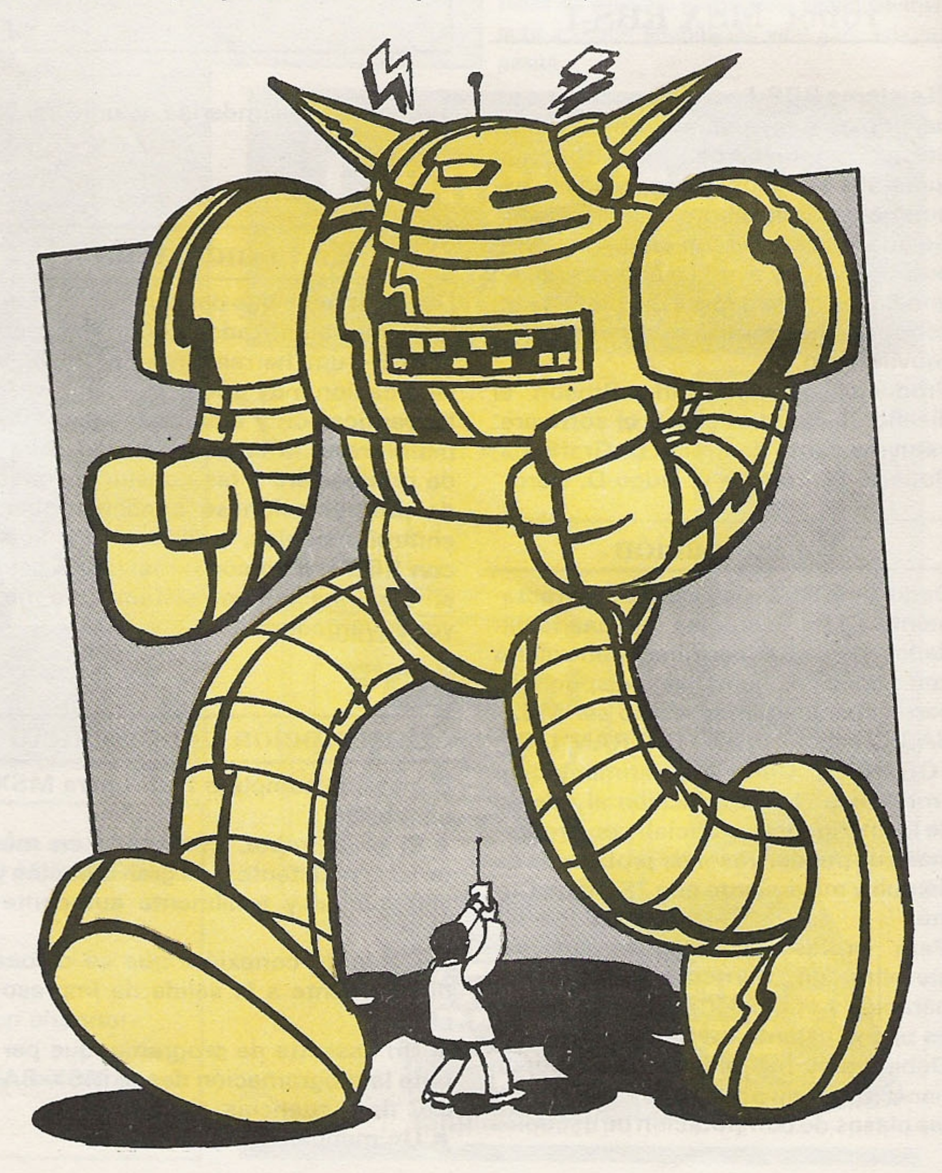
Podríamos profundizar en cada uno de los temas de la forma más teórica haciendo una exposición magnífica, o también tratar el asunto de la forma más empírica, pero de ninguna de las dos formas obtendríamos el resultado esperado: que lo entendamos todos.

Así que vamos a experimentar en cada una de las formas de manejo de archivos conocidas, de forma tal de llegar a algo concreto: la mayor aproximación posible de la comprensión.

El trabajo secuencial

Para empaparnos de esta clase de trabajo hemos creado un pequeño programa que se muestra en la figura 1, y sobre el cual basaremos nuestro pretencioso estudio.

Antes que nada deberemos copiarlo tal cual se encuentra en esa figura y grabarlo con el nombre que sugerimos y de la siguiente forma: SAVE "SECUEN 1.ASC",A.



De esta forma el programa quedará grabado en su forma ASCII. Que no es más que grabar el código ASCII de cada caracter que compone al listado incluyendo la tecla de RETURN o ENTER que aparentemente no figura en él; es decir, no produce ningún caracter visible.

Recomendamos que antes de copiar el listado se entre la instrucción sin número de línea: SIDTH 32, a los efectos de copiarlo sin errores ni modificaciones, pues hasta la longitud de los REMs tiene que ser la adecuada para que nos entendamos.

Por el momento nuestro objetivo se limita a tratar de crear un archivo numérico cualquiera y leerlo.

Entonces analicemos línea por línea lo que SECUEN1.ASC hará.

Le pusimos .ASC para diferenciarlo de ios demás programas que están seguramente grabados en forma comprimida por medio de tokens (números que simbolizan instrucciones BASIC).

En la línea 10 limpiamos la pantalla y la preparamos para trabajar a su máxima capacidad: 40 columnas.

Esto nos permitirá, en el momento de ejecutar la instrucción FILES, tener éste listado en tres columnas.

En los REMs de las líneas 20 y 30, el caracter pleno que allí figura es el que corresponde al de pulsar GRAPH P en la DPC-200, más conocida por TALENT MSX.

Debemos entonces crear un archivo secuencial que nos permita escribir datos numéricos en nuestro disco. Luego, en la línea 50 usamos la instrucción OPEN que abre un archivo seguido del nombre. Le indicamos al poner luego del FOR, que éste es de "salida" de datos por medio del OUTPUT y que es secuencial.

Pues, de no poner el FOR OUTPUT o INPUT la maquinola asume que trabajaremos en forma aleatoria.

Luego le indicamos que éste será el archivo número 1 gracias al AS#1. Esta cantidad (uno) es la única que nos permite tener sin usar una instrucción apropiada (MAXFILES) para aclarar que usaremos mayor cantidad de archivos al mismo tiempo.

Para escribir entonces, 100 números cualquiera bajo el nombre de AMADEVS, se usa el bloque de las líneas 50, 60 y 70.

Por medio de la línea 60 se grabarán números generados al "azar" entre 1 y 100.

Y en la línea 70, luego de terminar el bucle se cierra el archivo 1 por medio de la instrucción CLOSE #1. Avisándonos de ésto en la línea 80 y mostrándonos cómo quedó el directorio de nuestro disco.

Hasta aquí todo está muy lindo. Veamos entonces como quedó ésto grabado en el diskette.

En la línea 100 volvemos a habilitar el archivo AMADEVS pero esta vez como "entrada" de datos, difiriendo del anterior OPEN sólo en el comando FOR INPUT.

En la siguiente línea limpiamos la pantalla y creamos una variable (X) que nos ayudará a recordar el número de registro que estemos leyendo.

La línea 130 es la encargada de avisarnos y dar por terminada nuestra incursión en el disco si es que se ha terminado el archivo. Esto se logra gracias a la instrucción EOF (End Of File) que solamente requiere como parámetro al número del archivo que queremos que nos informe si terminó.

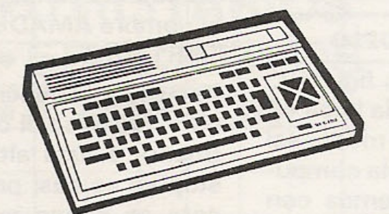
Así, si lo anterior no ocurre podemos leer un registro valiéndonos de la instrucción de la línea 140 que pondrá el valor numérico que encuentre en la variable A. Notemos, que si el valor buscado es numérico, no es por otro motivo que el de haber elegido una variable de estas características para acompañar al INPUT #1.

Hagamos pues correr este programa.

ROSARIO

MSX
LA NORMA INTERNACIONAL EN MICROCOMPUTACION

DPC 200



APLICACIONES PARA:
•LA EMPRESA
•EL HOGAR
•EL PROFESIONAL
•LA ESCUELA
•JUEGOS

CONTADO ₳ 495
PLANES DE FINANCIACION

MINICOMP S.R.L.
MAIPU 862 (2000) ROSARIO
Te: (041) 64-447
63-091 y 21-1266

ADEMAS DISPONEMOS DE:
•INSTITUTO DE ENSEÑANZA PROPIO
•LABORATORIO TECNICO PROPIO
•ASESORAMIENTO A ESCUELAS

Luego de escuchar por unos segundos el motor Mercedes de nuestro drive, veremos el mensaje que nos anuncia que ha terminado de grabar. Ahora sí podemos pulsar alguna tecla y ver que en cada uno de los 100 registros podemos leer los números tal cual fueron grabados.

Así que podemos usar esto en nuestros programas. Lo que no sabemos hasta el momento, es cuantos bytes en disco hemos ocupado con AMADEVS.

Comencemos pues a hacer las cosas prohibidas. En primer lugar cambiemos el nombre del archivo a leer de la línea 100 (AMADEVS) por el de SECUEN1.ASC.

Corramos a partir de la línea 100 esta nueva estructura, con GOTO 100.

Al referirnos a nuestro propios programa como un archivo secuencial, del que queremos leer simplemente valores numéricos, lo que aparecerá acompañando a cada registro será precisamente un caracter cuyo código ASCII tiene representación numérica.

En otras palabras, son todos los numeritos que se encuentran en nuestro listado Basic. Así podremos ver todos los números de línea de nuestro listado y los valores numéricos que en dichas líneas se encuentren.

Para ver otro efecto interesante, cambiemos además, en las líneas 140 y 150, la variable A por A$.

Corramos nuevamente el programa con GOTO 100, y veremos que los registros ahora contienen a las líneas de nuestro programa original. Entonces tendríamos que obtener de este tipo de lectura 17 registros, correspondientes a cada una de las líneas de nuestro SECUEN1.

Pero como las líneas 60 y 140, contienen como parte de sus instrucciones al caracter ",", ésto hace (por ser un delimitador de registros válido cuando se usa la instrucción INPUT# para lectura) que estas dos líneas ocupen dos registros cada una.

Volviendo a nuestro archivo original (AMADEVS), se presenta la inquietud de ver que es realmente, byte por byte, lo que quedó en el disco, y como es que guarda un valor numérico cuando es entrado por medio de un simple PRINT #.

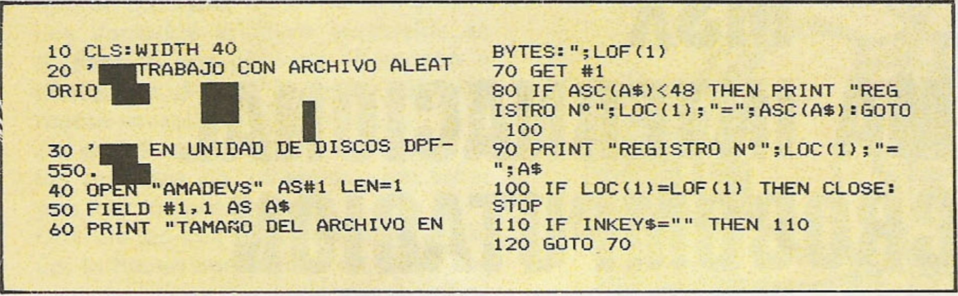
Para esto necesitamos alguna instrucción que nos permita desmenuzar adecuadamente un arreglo de esta naturaleza.

Y esto se puede hacer cómodamente usando una forma de lectura aleatoria. Así que...

Figura 1

```
10 CLS:WIDTH 40
20 '[redacted]TRABAJO CON ARCHIVO SECUE
NCIAL[redacted]
30 '[redacted] EN UNIDAD DE DISCOS DPF-
550.
40 OPEN "AMADEVS" FOR OUTPUT AS#
1
50 FOR F=1 TO 100
60 PRINT #1, INT(RND(1)*100)
70 NEXT F:CLOSE #1
80 FILES:PRINT:PRINT "YA HE GRAB
ADO 100 NUMEROS EN SU DISCO.  PUL
SE ALGUNA TECLA ([PAT])..........
..."
90 IF INKEY$="" THEN 90
100 OPEN "AMADEVS" FOR INPUT AS#
1
110 CLS:X=0
120 X=X+1
130 IF EOF(1) THEN PRINT "SE TER
MINO EL ARCHIVO.":CLOSE#1:END
140 INPUT #1,A
150 PRINT "El registro nº";X;"co
ntiene:";A
160 IF INKEY$="" THEN 160
170 GOTO 120
```

Figura 2



```
10 CLS:WIDTH 40
20 '[redacted]TRABAJO CON ARCHIVO ALEAT
ORIO[redacted]
30 '[redacted] EN UNIDAD DE DISCOS DPF-
550.
40 OPEN "AMADEVS" AS#1 LEN=1
50 FIELD #1,1 AS A$
60 PRINT "TAMAÑO DEL ARCHIVO EN
BYTES:";LOF(1)
70 GET #1
80 IF ASC(A$)<48 THEN PRINT "REG
ISTRO Nº";LOC(1);"=";ASC(A$):GOTO
 100
90 PRINT "REGISTRO Nº";LOC(1);"=
";A$
100 IF LOC(1)=LOF(1) THEN CLOSE:
STOP
110 IF INKEY$="" THEN 110
120 GOTO 70
```

El trabajo aleatorio

Copiemos el programa de la figura 2. En primer lugar el OPEN de la línea 40 carece de especificación de modo, así que, como decíamos antes, la computadora asume que trabajaremos con instrucciones de acceso aleatorio.

En esa misma línea se especifica que cada registro leído constará de un solo byte (LEN=1).

La siguiente línea se encarga de crear un campo o espacio de 1 byte llamado A$ dentro del buffer o espacio de memoria reservado para la comunicación entre la máquina y el drive.

En la línea 60, el comando LOF (Len Of File) nos da la longitud del archivo que figura en el número 1, en bytes.

Y ya por medio del GET #1 (línea 70), pasa el primer caracter del archivo a ese espacio (A$) que reservamos en el buffer.

Por medio de las siguientes dos líneas, diferenciamos los valores que en ASCII representan números de los que no, fijándonos si ellos son menores que 48 (valor ASCII que representa al 0) e imprimimos los valores que tengan significado numérico como tales y los que no, tal cual figuran en el disco.

La instrucción LOC (LOCates) que se usa en estas líneas, nos da el número de registro que estamos leyendo.

Como seguramente hemos notado, usando este tipo de lectura un registro puede tener la longitud que a nosotros nos sea más conveniente, esto es: no existen delimitadores más que el fin del archivo mismo.

La línea 100 es la encargada de verificar cuando se termina la lista de datos que estamos leyendo y así dar por concluída la ejecución del programa.

Al correr (RUN) entonces a ALEATOR1, veremos que el archivo consta de 593 bytes y no de 100 como era de esperar.

Si vemos detenidamente, el primer registro es el caracter 32 que corresponde al espacio, el segundo y el tercero corresponden al número 59, el cuarto a un espacio, el quinto al caracter 13 que es el token de RETURN o ENTER y el sexto al 10 que significa LINE FEED, que como caracter de impresora provoca que se deje un renglón en blanco antes de imprimir la siguiente línea.

Y esto se repite para cada número, así que para grabar cada uno de estos se utilizan 6 bytes en el disco.

Por supuesto sería mucho más económico, en cuanto al espacio utilizado, que cada número entero no mayor de 255, ocupará un byte, así el archivo sería seis veces menor.

Pero desgraciadamente, esto no es posible si utilizamos un diálogo secuencial para grabar este tipo de datos.

Por último veamos que es lo que ocurre cuando cambiamos en la línea 40, el nombre AMADEVS por el de ALEATOR1.ASC.

Borremos también la línea 80 y veamos qué pasa al correrlo.

Bueno, a esta altura el resultado de stop ya es casi predecible por todos, pero es bueno que experimentemos con estas cosas, para familiarizarnos con las instrucciones básicas que comandan a estos dos modos de trabajo con archivos.

Probemos entonces, cambiar por ejemplo, la longitud de los registros, el campo reservado para su lectura y escritura en el buffer o agregar otra variable para lectura, por supuesto refiriéndonos a ALEATOR1.

En fin, hagamos todos los cambios que seamos capaces de hacer, y experimentemos, que es una de las mejores formas de aprender, o por lo menos de adquirir confianza en este tema.

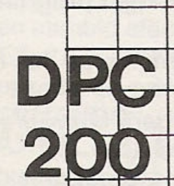
PORQUE LA COMPUTACION ES EL FUTURO...



MICROMATICA srl.

LOS PROFESIONALES DE LA COMPUTACION

ENTREGA INMEDIATA



DPC 200

SOFTWARE

- JUEGOS
- UTILITARIOS
- EDUCATIVOS
- A MEDIDA

HARDWARE

- EQUIPOS
- DISKETTERAS
- ACCESORIOS
- IMPRESORAS

BIBLIOGRAFIA - CURSOS (NIÑOS, ADULTOS, PROFESIONALES)

DISTRIBUIDORES OFICIALES

Talent MSX

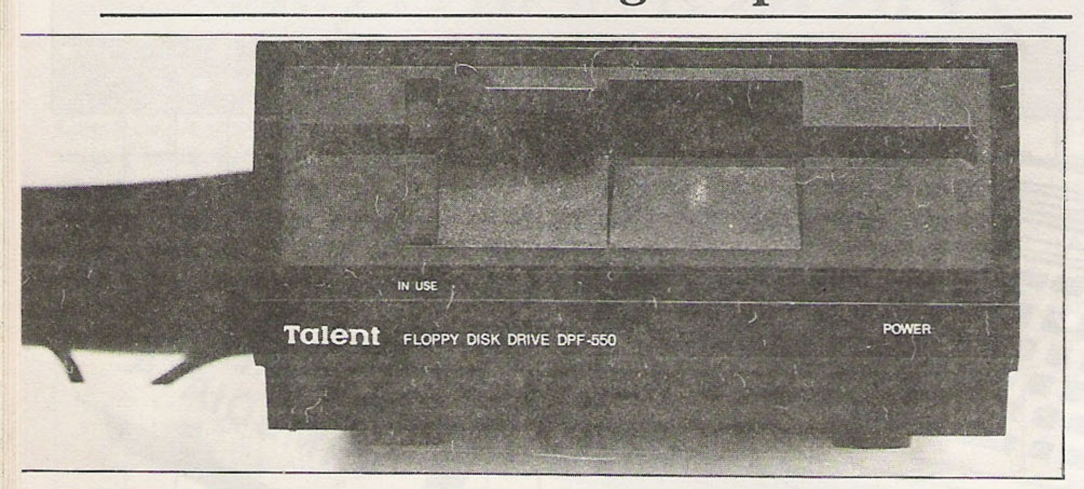
SPECTRAVIDEO

SVI

AV. PUEYRREDON 1135 (1118) Tel.: 821-5578

EL SISTEMA OPERATIVO

Continuamos viendo cada uno de los comandos que hacen del MSX-DOS una herramienta de gran potencia.



LINE INPUT#: Esta instrucción funciona en forma muy similar al INPUT#. De hecho, se la utiliza para leer un registro de 254 bytes o caracteres, de un archivo abierto para uso secuencial.

Su diferencia real radica en la forma en que esta instrucción nos permite leer dicho registro. Pues, el único delimitador reconocido por ésta es el llamado retorno de carro, más conocido como ENTER. Así esta instrucción leerá un registro (cuya máxima longitud puede ser de 254 caracteres) hasta que encuentre el token o byte que representa al ENTER, o bien finalice su capacidad.

Los parámetros que obligadamente deben acompañar a ésta son: en primer lugar un número que ha de coincidir con el del archivo abierto para este fin y en segundo lugar una variable alfanumérica cualquiera en donde se depositará lo que se haya leído del drive.

Un ejemplo, entonces, tendría la siguiente forma:

LINE INPUT#1,MARITO$

Si recordamos aquello que seguramente leímos en el manual de la máquina, que hablaba sobre las dos formas simples y posibles de grabar un programa basic, entenderemos la aplicación que sigue.

Por las dudas, recordemos que un programa puede ser grabado tanto en la forma en que la computadora lo almacena en memoria (forma que usamos comunmente, CSAVE) como en su representación ASCII. Esto es posible en cassette gracias a la instrucción SAVE (en su momento veremos como se hace ésto desde los comandos del DOS). De esta forma al grabar, tanto en disco como en cassette, un programa BASIC en su formato ASCII, cada línea o registro terminará con el caracter correspondiente al RETURN o ENTER.

Y esto nos permite pues leer un programa de esta naturaleza como si fuera un archivo secuencial, en el que cada registro correspondería a una línea BASIC. ¿Para qué nos puede servir esto?

Bien, una de las aplicaciones que se le puede dar, es la que surge de la necesidad de imprimir los listados a distintos anchos de columnas. Leyendo pues cada línea del programa que queremos listar por ejemplo en 10 columnas, podemos controlar la longitud del registro que saldrá por la impresora, simplemente cortándola en la variable que acompaña a LINE INPUT.

Esto justamente es lo que hace el programa "Formita.ASC" que acompaña a esta nota. Los caracteres especiales que acompañan a los LPRINT corresponden a los de una impresora Brother M-1009, pero seguramente nos daremos maña (si es que no coinciden) para buscar el reemplazo de éstos en la impresora de que dispongamos.

LOAD: Sobre esta instrucción no hay mucho que decir. Se la utiliza, como era de esperar, para cargar programas escritos en BASIC.

Veamos algunos ejemplos de sus distintas posibilidades.

LOAD "MEME": cargará el programa MEME desde el drive A.

LOAD "BRUJAS",R: cargará y ejecutará el programa BRUJAS del drive A.

LOAD "B:KDTS": cargará el programa KDTS del drive B, si es que tenemos la fortuna de poseer un segundo drive, o mejor dicho si tenemos la fortuna para poseer un segundo drive...

Es bueno notar que este comando nos permitirá leer desde un disco, tanto un programa grabado en formato "standard" como uno grabado en formato ASCII.

LSET y RSET: Antes de ver para qué se las utiliza, aclaremos algunas cosas.

Un archivo de acceso aleatorio tiene por supuesto distintas instrucciones de carga y lectura asociadas, a las de un archivo secuencial.

Las instrucciones (tanto de lectura como de escritura en disco) asociadas a los archivos de acceso aleatorio trabajan asociadas al llamado buffer o aguantadero provisorio de datos.

En este aguantadero, por ejemplo, se detienen a esperar los datos que queremos pasen al disco hasta que el drive diga: "estoy listo".

Pero este lugar o banco de memoria llamado buffer, no es una bolsa de papas en donde se tira la información a guardar en forma desordenada y a montones, como casi todo, tiene sus reglas.

Estas reglas son las que vimos cuando hablamos de la instrucción FIELD.

Veamos un ejemplo entonces, de lo

```
FIG1
10 UPEN "TIEMPO.DAT" AS#1
20 FIELD #1,10 AS A$
30 FOR F=1 TO 100
40 INPUT G
50 LSET A$=MKI$(G)
60 PUT #1,F
70 NEXT F
90 CLOSE #1
```

```
Formita.ASC
10 KEY OFF:CLEAR 600
20 CLS:FILES:PRINT:PRINT:PRINT:
PRINT
30 PRINT"Recordá que el program
a debe estar       grabado como <
NOMBRE.ASC>.":PRINT:PRINT:PRINT
40 MAXFILES=3:E$=CHR$(27):D$=E$
+"G":G$=E$+"E":GRAN$=CHR$(14):CH
ICO$=CHR$(15):NGRAN$=CHR$(20):NC
HICO$=CHR$(18)
50 LPRINT NCHICO$;NGRAN$;E$;"H"
;E$;"F";CHR$(13)+CHR$(10);E$;"A"
;CHR$(11);E$;"2"
60 LINE INPUT "Entre el nombre
del programa a listar   y el dis
co que lo contiene:";A$
70 IF A$="" THEN RUN
80 LPRINT D$+G$+GRAN$+A$;NGRAN$
+CHR$(13)+CHR$(10)+CHR$(10):A$=A
$+".ASC"
90 OPEN A$ FOR INPUT AS#1
100 IF EOF(1) THEN CLOSE #1:GOT
O 160
110 LINE INPUT#1,A$
120 FOR F=1 TO LEN(A$)
130 X=F/30:IF X-INT(X)=0 THEN L
PRINT CHR$(13)
140 LPRINT MID$(A$,F,1);
150 NEXT F:LPRINT CHR$(13)+CHR$
(10);:GOTO 100
160 LPRINT NCHICO$;NGRAN$;E$;"H
";E$;"F"
```

que hacen estos dos comandos.
Supongamos que hemos definido un campo en el buffer llamado A$, cuya longitud es de 20 lugares, y la palabra que queremos guardar tiene una longitud menor que la reservada, por ejemplo "ANDI", entonces estas instrucciones se comportarán como sigue: LSET A$="ANDI" llenará con espacios los siguientes 16 lugares a ANDI, y RSET A$="ANDI" cargará en A$ 16 espacios en blanco y luego completará los siguientes 4 lugares con "ANDI". Este proceso se llama justificar a derecha o a izquierda.
También puede usárselas en combinación con la impresora o la pantalla, pero su funcionamiento es levemente distinto, veamos.
Supongamos que en la variable CRISTIAN$ tenemos almacenados 30 espacios, si efectuamos la instrucción: RSET CRISTIAN$="FERNANDO" y luego imprimimos CRISTIAN$ por pantalla o impresora, veremos que se dejan los 22 espacios libres correspondientes al resto de los 30 y luego la palabra FERNANDO.
Si en cambio efectuamos la instrucción LSET CRISTIAN$="FERNANDO" se imprimirá en primer lugar la palabrita "FERNANDO" y luego los 16 espacios.

MAXFILES: Esta es una de las variables peculiares del DOS.
En ella se "almacenará" el valor numérico correspondiente a la cantidad máxima de archivos que usaremos a la vez en un programa nuestro, tanto de lectura como de escritura.
El rango de valores posibles de MAXFILES es de 0 a 15.
Si efectuamos, por ejemplo: MAXFILES=12 podremos usar a la vez esa cantidad de archivos.
Si el valor de esta bendita variable es cero, sólo podremos usar las instrucciones SAVE y LOAD, para manejar archivos.
Si no especificamos otra cosa, normalmente, desde la iniciación del sistema, MAXFILES tendrá el valor 1, con el consabido efecto.

MERGE: Se usa para agregar un programa BASIC grabado en ASCII y residente en disco, al programa BASIC, también, residente en el momento de llamarlo en la memoria de la máquina.
Si alguno de los números de línea del programa que tenemos en memoria coincide con el del programa que se está mezclando desde disco, esta última no será cargada.
Como ejemplo podríamos ejecutar lo siguiente: MERGE "GONZALO" siempre y cuando el programa "GONZALO" esté grabado en el disco en forma ASCII.

NAME: Sirve para cambiarle el nombre a archivos existentes en el directorio de nuestro diskette.
Así, NAME "CLAUDIO" AS "CORTAZAR" le cambiará el nombre al archivo existente en el disco, cuyo nombre era CLAUDIO por el de CORTAZAR.
El programa o archivo en sí no es tocado en absoluto, sólo su nombre.

OPEN: Esta sentencia permite interactuar con la diskettera de varias formas y con varios tipos de archivos.
La forma general que toma ésta junto a sus parámetros es:
OPEN "nombre del archivo" FOR modo AS # número asignado LEN=longitud.
Donde "modo" corresponde a la palabra que identificará al tipo de archivo que por medio de esta instrucción se está habilitando, y que puede ser: OUTPUT, INPUT, APPEND o simplemente nada.
El modo OUTPUT define al archivo en cuestión, como de salida de datos que serán grabados unos tras otros en forma secuencial.
El modo INPUT define al archivo de la misma forma que el anterior pero para entrada de datos desde la unidad de discos.
El modo APPEND al igual que los anteriores define al archivo como secuencial y permite añadir registros a un archivo de esta clase ya existente en el disco.
Si simplemente se omite la frase "FOR modo", la computadora asume que el archivo que se está definiendo será tanto de lectura como de escritura en disco y que el acceso será aleatorio.
El número que figura luego del símbolo numeral identificará al archivo, y debe estar comprendido en el rango que se haya especificado en la variable "MAXFILES".
Por último, 'longitud' es un número entero, que de ser puesto junto a 'LEN=' indica la longitud que tendrán los registros.
Este puede variar entre 1 y 256. Sólo se lo puede utilizar en archivo del tipo de acceso aleatorio.
Debemos tener en cuenta que si abrimos un archivo para salida de datos en forma secuencial, y existía otro con el mismo nombre, este último será destruido.
Otra de las facilidades es la de poder tener un mismo archivo abierto para lectura en forma secuencial con un determinado número y, con otro número, tener abierto a la vez este mismo en forma aleatoria.

PRINT# y PRINT# USING: Tanto la primer instrucción como la segunda, se utilizan para enviar datos a la unidad de discos en forma secuencial.
Veamos con ejemplos como trabajar con ella:
PRINT #1,OSCAR;PATRICIA grabará en el archivo número 1 del disco el contenido de las variables OSCAR y PATRICIA.
PRINT #, OSCAR$;PATRICIA$ grabará en el archivo correspondiente el contenido de esas dos variables alfanuméricas. Si, OSCAR$="Editorial" y PATRICIA$="PROEDI" en el disco la información quedará como Editorial PROEDI.
Así, cuando leamos por medio de la instrucción INPUT#1,A$ en esta variable quedará almacenado Editorial PROEDI.
Si en cambio OSCAR$="Editorial," y PATRICIA$="PROEDI" en el disco quedará lo siguiente: Editorial, PROEDI.
Luego cuando leamos por primera vez con INPUT #1,A$ en esta última quedará "Editorial" y al hacerlo por segunda vez quedará PROEDI, pues la coma es un delimitador válido.
El comando adicional USINT se puede utilizar al igual que en pantalla para formatear los datos a salir por impresora.

PUT: Este es el comando que nos permitirá colocar un dato en el buffer, para luego ser mandado al disco en el archivo de acceso aleatorio que se ha abierto para tal fin.
En la figura 1 vemos un ejemplo de aplicación de esta instrucción. A esta altura de los acontecimientos ya podemos comprender su funcionamiento perfectamente, sin más aclaraciones.

RUN: Este comando seguido del nombre de un programa válido, hará que éste se cargue en memoria y se ejecute automáticamente.
Esta instrucción cierra todos los archivos que hasta el momento de ejecutarla, hayan permanecido abiertos. Pero si agregamos una 'R' luego del nombre, éstos permanecerán abiertos.
De esta forma, RUN "TAMI",R dejará sin tocar los archivos que se hayan abierto hasta el momento, cargará el programa "TAMI" y por último lo ejecutará.

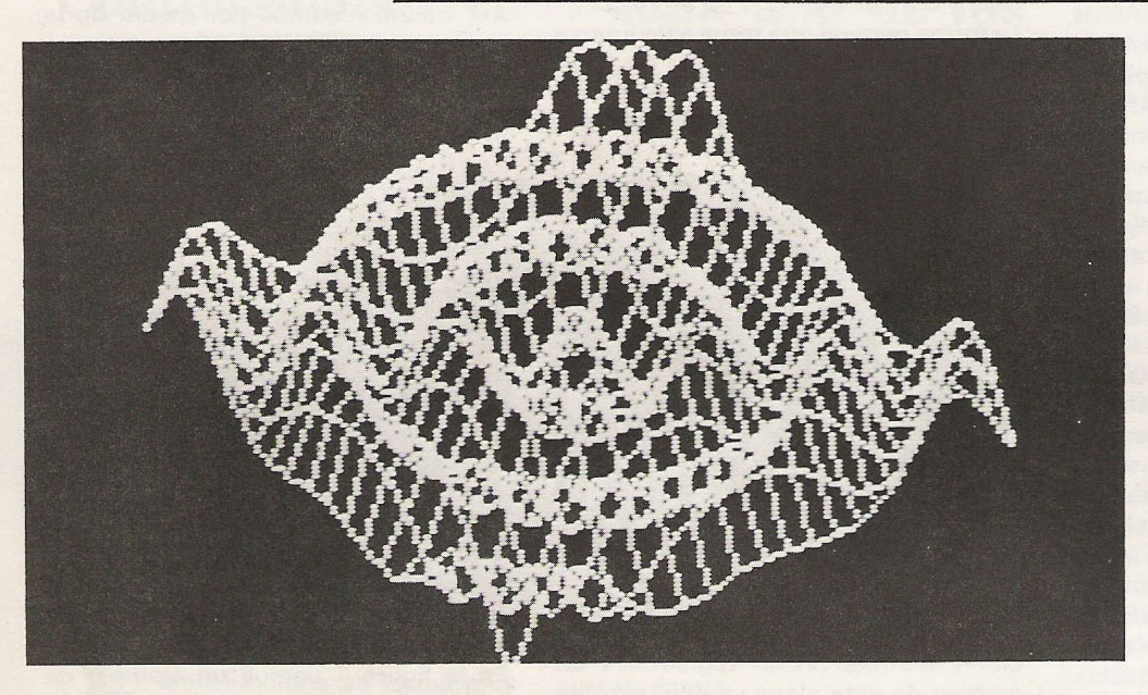
SAVE: ¿A que no saben lo que hace esta instrucción? ¿Cómo se dieron cuenta? Así no me dan ganas de contarle más nada.
Lo que a lo mejor no se han dado cuenta es que si luego del nombre que le dimos al programa para grabarlo, le agregamos una A, el programa será grabado en su versión ASCII.
Al grabar de esta forma, lógicamente se tarda más tiempo.

SYSTEM: Este es el nombre de la rutina que nos permitirá entrar en un modo editor de DOS, dentro del cual se encuentran las instrucciones que veremos en el siguiente número.
Dentro de la siguiente entrega veremos también algunos comanditos que se pueden usar también desde el intérprete BASIC.



JUGANDO EN TRES DIMENSIONES

Clase: educativo



Una de las aplicaciones más interesantes en máquinas dotadas de una capacidad gráfica considerable, es la exploración de la tercera dimensión.
Los gráficos en 3-D, además de su utilización práctica, logran sensaciones estéticas sumamente atractivas.

Puede volverse un juego el hecho de crear nuevas formas o vistas modificando parámetros y funciones, incluso sin introducirse en la búsqueda de nuevos métodos de perspectiva.

Para los que siempre están a la búsqueda de nuevas emociones, o para los que quieren comenzar a experimentar con formas nuevas de dibujar con la computadora, les ofrecemos aquí dos graficadores muy especiales.
El primero muestra una perspectiva isométrica de la función especificada en el programa, en base a dos parámetros de entrada: "XMAX" e "YMAX", que determinan el rango en que será evaluada la función. Se evalúa entre + - el valor ingresado (por ej: Si en "XMAX" se ingresa 5, se evalúa entre 5 y -5).
En primer lugar, la función es calculada y graficada simultáneamente en un sentido, y luego se barre en el otro, con los valores ya calculados.
Podemos modificar los ángulos de los ejes de la perspectiva alterando en la línea 20 los valores de "AY" y "AX", que van expresados en radianes.
Para cambiar la función debemos escribir la nueva a partir de la línea 2000, con el resultado en "Z", usando como variables "X" e "Y", y finalizando con un RETURN
Es muy importante que el valor de la función en todo punto a evaluar no sobrepase -15 o + 15 y no sea demasiado pequeño porque no se vería con claridad.
De todos modos experimenten cambiando todo lo que quieran, aunque los resultados puedan ser catastróficos. Ahí es donde está la verdadera diversión.
El segundo programa utiliza un método de visualización muy particular, ya que aquí no existen perspectivas, sino que la función está vista desde arriba, y las diferencias de altura están indicadas por distintos colores.
La forma de suministrarle los rangos y la función es la misma que en el caso anterior, pero el valor de la función no debe estar fuera del rango de 1 a 15,99.
Aun cuando la función y los límites sean los mismos, notarán que el efecto logrado es totalmente distinto en cada uno de los programitas.
Jueguen y diviértanse con esto, mientras nosotros desarrollamos nuevos entrenimientos.

Fabian Jofré
y Martin Salias
(Club de Usuarios)

```
10 COLOR 15,4,4
20 DIM GX(21,40),GY(21,40)
30 AY=.9:AX=.5:SX=SIN(AX):CX=COS
   (AX):SY=SIN(AY):CY=COS(AY)
40 INPUT "XMAX";B
50 INPUT "YMAX";D
60 A=-B:C=-D:EX=50/B:EY=50/D
70 SCREEN 2
80 FOR X=A TO B STEP (B-A)/20
90 Y=C
100 GOSUB 2000
110 GOSUB 1000
120 PSET(XX,YY)
130 RX=RX+1
140 RY=0
150 FOR Y=C TO D STEP (D-C)/40
160 GOSUB 2000
170 GOSUB 1000
180 GX(RX,RY)=XX
190 GY(RX,RY)=YY
200 RY=RY+1
210 LINE -(XX,YY)
220 NEXT Y,X
230 FOR Y=0 TO 40 STEP 2
240 PSET (GX(1,Y),GY(1,Y))
250 FOR X=1 TO 21
260 LINE -(GX(X,Y),GY(X,Y))
270 NEXT X,Y
280 GOTO 280
1000 YY=Z-SX*X*EX-SY*Y*EY+87
1010 XX=CX*X*EX-CY*Y*EY+127
1020 RETURN
2000 Z=-COS (SQR(X*X+Y*Y))*10
2010 RETURN
```

```
10 COLOR 15,1,1
20 CLS
30 DIM CO(15)
40 FOR A=1 TO 15:READ CO(A):NEXT
50 INPUT "XMAX";B
60 INPUT "YMAX";D
70 A=-B:C=-D:EX=50/B:EY=50/D
80 SCREEN 3:RY=-4
90 FOR Y=C TO D STEP (D-C)/192*4
100 RY=RY+4
110 RX=0
120 FOR X=A TO B STEP (B-A)/64
130 GOSUB 1000
140 PSET  (RX*4,RY),CO(Z)
150 RX=RX+1
160 NEXT X,Y
170 GOTO 170
180 DATA 1,4,12,6,13,5,8,2,3,7,
     9,10,11,14,15
1000 Z=COS (SQR(X*X+Y*Y))*7+8
1010 RETURN
```

```
Z=COS (X+Y)*7+8
Z=COS (SQR(X*X*X-Y*Y))*7+8
Q=SQR(X*X+Y*Y):Z=COS(Q)/Q*7+8
```

A LA COMPUTADORA PERSONAL

Talent MSX

NADA LE ES IMPOSIBLE ...



... y para que todo le sea posible a Ud. en:

- provisión del hardware tanto de la consola como todos los periféricos y accesorios.
- la instalación y la puesta en marcha.
- soft utilitario
- soft para la educación
- soft comercial

Se han asociado para brindarle una red de solidaridad y atención, los siguientes concesionarios amigos:

AVELLANEDA	**ARGOS S.A.** Avda. Mitre 1755 (CP 1870) Tel. 203-5227
CAPITAL	**COMPUTRONIC S.A.** Viamonte 2096 (CP 1056) Tel. 46-6185
CASTELAR	**HOT BIT COMPUTACION** Carlos Casares 997 (CP 1712) Tel. 629-2247
SAN ISIDRO	**F. CORATELLA S.R.L.** Cosme Beccar 249 (CP 1642) Te. 743-0734

EL MODEM TMX-510 DE TALENT

Nuevamente la empresa Telemática nos sorprende con un importante desarrollo: el Modem TMX-510. Veremos algunas de las características de este nuevo producto.

Este novísimo periférico nos permitirá desarrollar una de las más gratas actividades en materia informática: la Telemática.

Podremos comunicarnosc con él con otras máquinas o terminales, o con base de datos nacionales o extranjeros, a través de una línea telefónica común.

Las señales digitales que genera nuestra computadora, son transformadas, gracias a éste, en analógicas que son envueltas por una portadora modulada en frecuencia.

En el otro extremo de la línea es necesaria la presencia de otro modem que se encargue de traducir esta cola de datos codificados analógicamente en señales digitales.

el diálogo entre dos modems puede establecerse de dos formas:

Half duplex (transmisión de datos en un sentido a la vez).

Full duplex (transmisión de los mismos en dos sentidos a la vez).

En el primero de estos modos, antes de transmitir un dato, el modem "transmisor" debe verificar que el "receptor" esté en modo "escucha" y luego transmitir.

Es necesario pues que se den inversas condiciones, para lograr la comunicación en sentido opuesto.

La forma full duplex permite que la transmisión y recepción sean simultáneas. Usando para ello portadoras de distinta frecuencia para que no se produzca errores.

Para que este tipo de comunicación sea posible es menester que alguno de los equipos sea definido como ORIGINATE (que origina) y el otro como ANSWER (que responde).

Es bueno recordar que TALENT MSX responde a las normas CCITT y BELL que se han adoptado mundialmente para las transmisión de datos entre equipos de esta naturaleza.

Las velocidades de transmisión disponibles son las que se ven en la figura 1.

FIGURA 1
300 BAUDIOS norma CCITT ORIGINATE FULL DUPLEX
300 BAUDIOS norma CCITT ANSWER FULL DUPLEX
300 BAUDIOS norma BELL ORIGINATE FULL DUPLEX
300 BAUDIOS norma BELL ANSWER FULL DUPLEX
1200 BAUDIOS norma CCITT HALF DUPLEX

La memoria del modem

Esta caja mágica posee tres bancos de memoria de lectura solamente (EPROM) de 16 Kbyte cada uno y un cuarto chip de las mismas características de 32 Kbytes.

En ellos se almacena todo el software necesario para que la comunicación sea posible y todo lo necesario para agregar las nuevas instrucciones BASIC que comandarán a esta cajita.

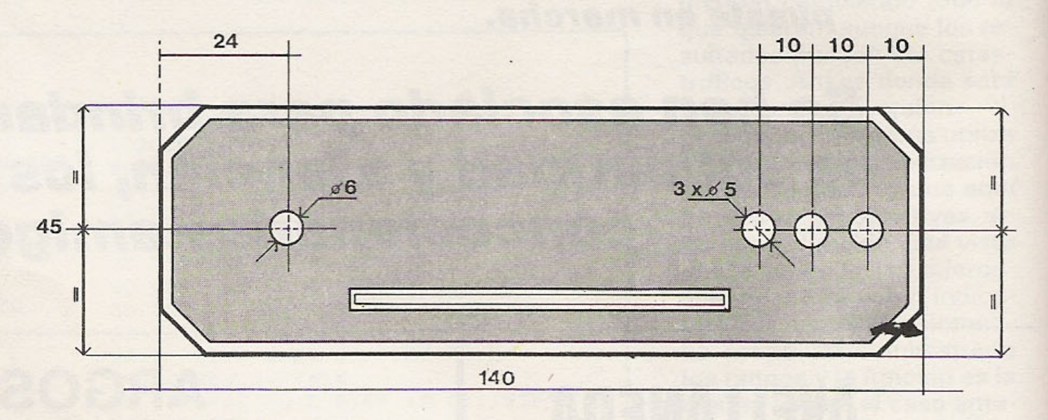
Al encender nuestra MSX, estando conectada a ella, se inicializará de forma tal que queda lista para acceder a este sistema.

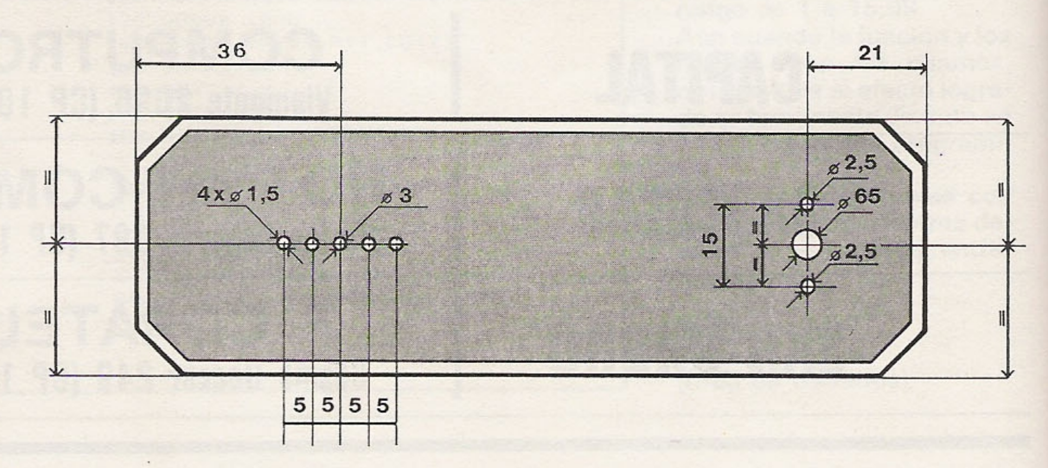
El canal de comunicaciones.

Este se abre por medio de la instrucción OPEN de la forma que sigue:

OPEN "COM(n):" (FOR (modo) AS ( ) (n: número de archivo))

Así, ésta abre un canal de comunicaciones para su posterior procesamiento, definiendo un campo el buffer (bloque de memoria destinado para





tal fin) de 128 bytes y prendiendo la portadora del modem.
"Modo" define si es salida secuencial o entrada secuencial. Si este "modo" no es especificado, el modem queda preparado para trabajar en ambos sentidos al mismo tiempo.
El número de archivo representa lo mismo que en cualquier otro archivo.
Entre las instrucciones nuevas que el uso de éste requiere, se encuentra:
SAVE "COMn:", a
que permite enviar un programa BASIC a través de la línea. Si al final de la instrucción se especifica el comando "a" éste será enviado en su forma ASCII. De otra forma el programa viajará tal cual se encuentra en memoria (instrucciones en tokens).
Aclaramos que "n" es el número de canal asignado a la transmisión.
Lo que no es posible enviar es el nombre del programa como tal.
También están disponibles las instrucciones LOAD y MERGE, que funcionan de forma similar.
Los que siguen son algunas de las llamadas (CALLs) a subrutinas internas:

CALL COMINI: inicializa los parámetros que regirán al modem.

CALL COMON: habilita la generación de una interrupción cuando es recibido un caracter del canal de comunicaciones especificado.

CALL COMOFF: deshabilita las interrupciones que validaba CALL COMON.

CALL COMSTOP: deshabilita la posibilidad de guardar cada carácter recibido en el buffer de comunicaciones indicado.

CALL COM: define la línea de comienzo de la subrutina hacia donde se producirá el salto del programa, cuando se genere una interrupción debido a la llegada de un carácter por el canal de comunicaciones.

CALL COMBREAK: envía un carácter que significa en el lenguaje de los modems, detener la transmisión.

CALL COMTERM: este comando permite emular una terminal de comunicaciones.

Existen algunas más, pero creemos que con estas basta para mostrar lo que es capaz de realizar esta cajita mágica.
Este es otro de los pequeños pasos que permiten lograr grandes cosas.
No puede ser de otra forma, pues este desarrollo mejora y activa la saludable costumbre de la cultura, acortando las distancias de la divulgación.
Por último, en la figura vemos un plano de las dimensiones físicas de este aparato.







Matemática MSX

Estas son las fórmulas de las funciones matemáticas más usadas.

Secante
SEC (X)=1/COS(X)

Consecante
CSC(X)=1/SIN(X)

Cotangente
COT(X)=1/TAN(X)

Arco seno
ARCSIN(X)=ATN(X/SQR(–X*X+1))

Arco coseno
ARCCOS(X)=–ATN(X/SQR(–X*X+1))+1.5708

Arco secante
ARCSEC(X)=ATN(X/SQR(X*X–1))+SGN(X)–)*1.5708

Arco cosecante
ARCCSC(X)=ATN(X/SQR(X*X–1))+(SGN(X)–1)*1.5708

Arco cotangente
ARCCOT(X) = =ATN(X)+1.5708

Seno hiperbólico
SINH(X)=(EXP(X)–EXP(–X))/2

Coseno hiperbólico
COSH(X)=(EXP(X)+EXP(–X))/2

Tangente hiperbólica
TANH(X)=(EXP(X)–EXP(–X))/EXP(X)+EXP(–X))

Secante hiperbólica
SECH(X)=2/(EXP(X)+EXP(–X))

Secante hiperbólica
CSCH(X)=2/(EXP(X)–EXP(–X))

Cotangente hiperbólica
COTH(X)=(EXP(X)+EXP(–X))/(EXP(–X))

Arco seno hiperbólico
ARC- SINH(X)= =LOG(X+SQR(X*X+1))

Arco coseno hiperbólico
ARCCOSH(X)=LOG(X+SQR(X*X–1))

Arco tangente hiperbólica
ARCTANH(X)=LOG((1+X)/(I–X))/2

Arco secante hiperbólica
ARCSECH(X)=LOG((SQR(–X*X+1)/X)

Arco secante hiperbólica
ARCCSCH(X)=LOG((SGN(X)*SQR(X*X+1)+1)/X)

Arco cotangente hiperbólica
ARCCOTH(X)=LOG((X+1)/(X–1))/2

Títulos y explicaciones

Los comentarios que se utilizan en los programas, sobre el manejo y utilidad de los mismos, aparecen de una forma muy tradicional. Generalmente son frases que de golpe se ven sobre la pantalla.

Pero es hora que rompamos con las formalidades de presentación de los programas. Por eso aquí proponemos dos formas un poco más originales. Una de ellas coincide en hacer imprimir los comentarios y aclaraciones necesarios de un programa como si estuvieran desfilando a lo ancho de la pantalla. Pero si bien es vistosa, es un poco dificultosa para leer.

Esto se soluciona incrementando el tiempo de espera hasta que aparezca la próxima letra.

En el programa de la figura uno, el FOR de la línea setenta es el resultado para permitir leer las palabras. Si necesitamos más tiempo para poder leer, se puede cambiar el tope 30 de esa línea por cualquier otro valor. De esta manera conseguiremos que las palabras no huyan de nuestra visual.

Otra manera original de hacer presentar las explicaciones es que suban desde la parte inferior de la pantalla, de una forma muy similar a la que se suele ver cuando las películas anuncian a los actores. En la línea 60 del programa de la figura dos, se realiza el centrado de la cadena a imprimir.

En la línea 30 se declara la pantalla de 40 caracteres de ancho. Por este motivo el razonamiento para hacer centrar la cadena es dividir la longitud de la misma por la mitad y restárselo a la mitad de la pantalla. El resultado sería entonces la cantidad de caracteres libres sobre el margen izquierdo. Luego se imprime la variable alfanumérica.

Figura 2

```
10 " ** LOAD MSX **"
20 SCREEN 0
30 WIDTH 40
40 FOR I=1 TO 4
50 READ A$
60 LOCATE 20-(LEN(A$)/2),24
70 PRINT A$:PRINT CHR$(10):
   PRINT
80 NEXT I
90 DATA ESTO,NO,PEUDE,SEGUIR
```

Figura 1

```
10 REM ** LOAD MSX**
20 CLS:COLOR1,7: SCREENO:WIDTH 40
30 A$="PANTALLA UTILIZANDO LAS POSIBILIDADES
   DE MID$ ESTO ES UN DESFILE DE TEXTO'
   EN LA PANTALLA UTILIZANDO LAS POSIBILIDADES
   DEL MID$"
40 P=41
50 LOCATE 0,10
60 PRINT MID$(A$,P,40)
70 FOR F=1 TO 30:NEXT
80 P=P+1
90 IF P=LEN(A$)-41 THEN P=1
100 GOTO 50
```

En el DATA de la línea 90, se pueden agregar tantas palabras como sean necesarias, pero la cantidad de cadenas de caracteres debe coincidir con el tope del FOR de la línea 40.

Si este número es menor que la cantidad de datos a leer, sólo aparecerá en la pantalla el número de veces que se incremente el FOR. Pero si el número del FOR supera al de datos, aparecerá en la pantalla un mensaje de error.

MSX es un cuentavuelta

Para aquellos que tengan grabador sin contador, les proponemos un método para encontrar un programa y usar la computadora como contador.

Antes de introducir la instrucción SAVE, inicialicemos el reloj interno de la computadora en cero y después sí ya podemos entrar la instrucción para grabar.

Al terminar de salvar el programa en cinta, pidamos que imprima el valor TIME. Con esto sabremos cuánto tiempo dura el pasaje del programa por el grabador. No hace falta introducir estas instrucciones con un número de línea, basta con entrarlas de la siguiente manera:

```
TIME=0: CSAVE "nombre":
PRINT TIME
```

Para cargar un programa, usemos la siguiente subrutina:

```
65000 TIME=0: MOTOR ON
65010 IF TIME=num THEN MOTOR OFF: GOTO 65040
65020 GOTO 65010
65030 CLOAD "nombre":
PRINT TIME
```

donde num equivale al tiempo que debemos dejar correr la cinta antes de leer algún programa. "nombre" responde al programa que queremos leer.

Bien aleatorios

Si bien es cierto que la función RND genera números aleatorios, también es verdad que siempre es la misma secuencia de números. Pero si conseguimos introducir la variable TIME, al ser ésta una variable que cambia de valor 50 veces por segundo, nos permitirá obtener verdaderos valores al azar.

Por ejemplo X=INT (RND (TIME)*A)+1

Esta instrucción genera valores entre 1 y A.

Talent MSX

DISTRIBUIDORES OFICIALES

COMPUPRANDO S.C.A.
Av. de Mayo 965
(1085) Capital
Te.: 38-0295

COMPUSHOP S.A.
Córdoba 1464
(1055) Capital
Te.: 41-8730 - 42-9568
49-2165

ARGECINT S.R.L.
Ventura Bosch 7065
(1408) Capital
Te.: 641-0327/4892
641-3051

CP 67 CLUB S.A.
Florida 683 Loc. 18
(1005) Capital
Te.: 393-6303

MICROSTAR S.A.
Callao 462
(1022) Capital
Te.: 45-0964/1662

SIST. LOGICAL S.R.L.
Esmeralda 561 3º B
(1007) Capital
Te.: 393-7669

(AREA EDUCATIVA
UNICAMENTE)

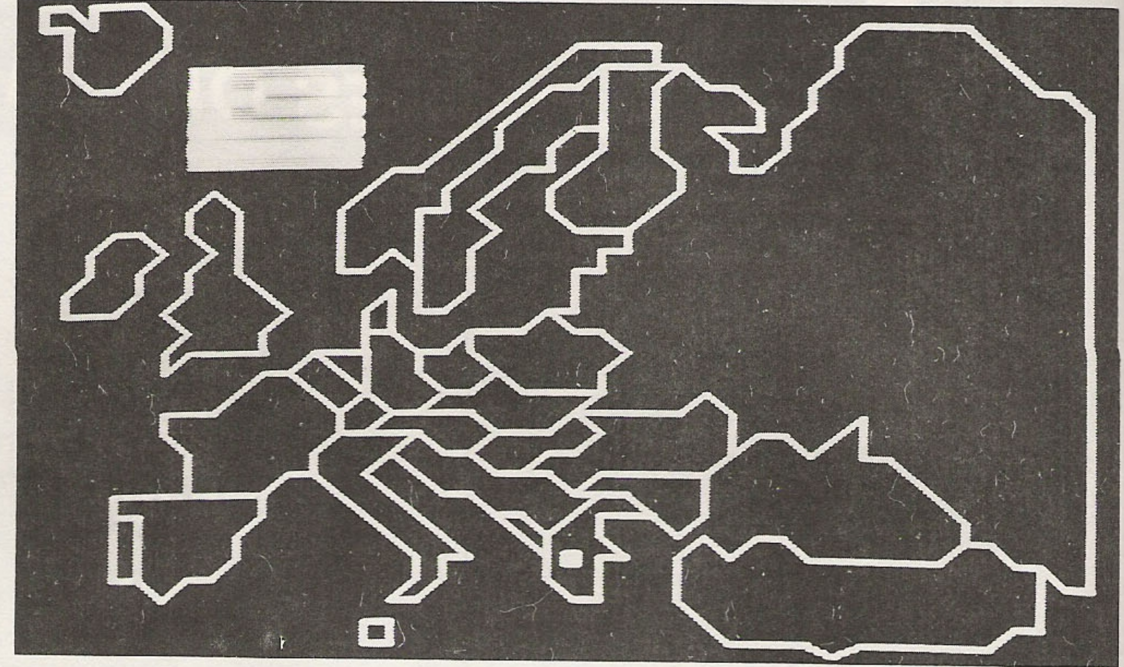
DIST. CONCALES S.A.
Tucumán 1458
(1050) Capital
Te.: 40-8664/0344

MICROMATICA S.R.L.
Av. Pueyrredón 1135
(1118) Capital
Te.: 821-5578



EUROPA

clase: educativo



Este programa es de gran utilidad para aquellos que deban estudiar las capitales de los países de Europa.
La tarea puede resultar aburrida para los que deban recurrir a tediosos libros. Pero este es un juego tan divertido y ameno que nos permitirá desafiar a cualquier profesor hasta que seguramente lo derrotaremos en alguna capital (como la de Islandia).
El juego no sólo nos informa sobre una ciudad sino que también nos muestra la ubicación geográfica de la misma en el mapa. Así, primero nos preguntará por la capital, nos dirá si supimos contestar y en caso contrario nos corregirá.
Además, nuestra maquinita se siente lo suficientemente inteligente y no cree que conozcamos la bandera de dicho país. Por lo que tratará de humillarnos mostrándonos el símbolo patrio de cada país.
La aparición de los países está dada por la función RND y esto hace que algunas veces se repita más de una vez la misma pregunta. Dejamos para ustedes la opción de modificar esto y como sugerencia les aconsejamos guardar el número del país elegido en una matriz auxiliar de dimensión 28. Aunque vale aclarar que los números ya están guardados en una matríz. Cada vez que se fije un número como posible país a interrogar, se verifica que dicho número no se encuentre en la matriz auxiliar. Si el número ya había salido se preguntará por otro hasta obtener uno distinto a todos los aparecidos.
Otra variante posible sería agregar una variable que cuente las cantidades de países que se nos preguntó y en otra variable la cantidad de capitales que supimos contestar correctamente.

Estructura del programa:

30-110 inicializa la matríz con los países y otra para las capitales
120-160 pantalla inicial de introducción
170-190 elige modo tres de pantalla y escribe Europa bien grande sin tener que definir caracteres gráficos (este es un ejemplo del uso del modo tres)
200-410 pregunta por la capital y verifica la respuesta, acompañado por mensaje aclaratorio y música de derrota o de triunfo, según haya sido la contestación.
420-630 dibuja el mapa
640-2590 grafica la bandera del país en cuestión y nos marca en el mapa la ubicación geográfica del mismo.

Variables importantes

N$: matriz con los países
C$: matriz con las capitales
X : subíndice de la matriz con los países que fue elegido por el RND en la línea 230

```
30 DIM N$(28),C$(28)

40 N$(1)="ALBANIA":N$(2)="AUSTRIA":N$(3)="BELGICA":N$(4)="BULGARIA":N$(5)="CHECOLOVAQUIA":N$(6)="DINAMARCA":N$(7)="ESPA"+CHR$(&HA5)+
"A":N$(8)="FINLANDIA"

50 N$(9)="FRANCIA":N$(10)="GRAN BRETA"+CHR$(&HA5)+"A":N$(11)="GRECIA":N$(12)="HOLANDA":N$(13)="HUNGRIA":N$(14)="IRLANDA":N$(15)="ISL
ANDIA":N$(16)="ITALIA"

60 N$(17)="MALTA":N$(18)="NORUEGA":N$(19)="POLONIA":N$(20)="PORTUGAL":N$(21)="RD. ALEMANIA":N$(22)="RF. ALEMANIA":N$(23)="RUMANIA"

70 N$(24)="SUECIA":N$(25)="SUIZA":N$(26)="TURQUIA":N$(27)="U.R.S.S.":N$(28)="YUGOSLAVIA"

80 C$(1)="TIRANA":C$(2)="VIENA":C$(3)="BRUSELAS":C$(4)="SOFIA":C$(5)="PRAGA":C$(6)="COPENHAGUE":C$(7)="MADRID":C$(8)="HELSINKI":C$(9
)="PARIS"

90 C$(10)="LONDRES":C$(11)="ATENAS":C$(12)="AMSTERDAM":C$(13)="BUDAPEST":C$(14)="DUBLIN":C$(15)="REYKJAVIK":C$(16)="ROMA":C$(17)="LA
 VALLETTA"

100 C$(18)="OSLO":C$(19)="VARSOVIA":C$(20)="LISBOA":C$(21)="BERLIN":C$(22)="BONN":C$(23)="BUCAREST":C$(24)="ESTOCOLMO":C$(25)="BERNA
```




```
110 C$(26)="ANKARA":C$(27)="MOSCU":C$(28)="BELGRADO"
120 CLS:PRINT" INTRODUCIR LA CAPITAL DE LA NACION":PRINT"CORRESPONDIENTE"
130 PRINT:PRINT"Y PRESIONAR LA TECLA -RETURN-"
140 PRINT:PRINT"PARA TERMINAR EL PROGRAMA PULSAR LAS TECLAS CRTL-STOP A LA VEZ"
150 PRINT:PRINT:PRINT"PARA COMENZAR PRESIONAR CUALQUIER":PRINT "TECLA"
160 IF INKEY$="" THEN GOTO 160
170 OPEN "GRP:" AS#1 :SCREEN 3
180 PRESET (40,60)
190 PRINT#1,"EUROPA":FOR I=1 TO 2000:NEXT
200 CLS
210 SCREEN 0
220 KEY OFF
230 X=RND(-TIME):X=INT(RND(1)*28)+1
240 CLS
250 COLOR 15,4,4
260 LOCATE 8,10
270 PRINT"CAPITAL DE ";N$(X)
280 LOCATE 8,12
290 INPUT R$
300 IF R$=C$(X) THEN GOTO 380
310 PRINT" 0000000 FALLO 0000000"
320 LOCATE 8,16
330 PRINT "ES ";C$(X)
340 PLAY "CCCCCCC"
350 FOR I=1 TO 2000:NEXT
360 CLS
370 GOTO 440
380 PRINT"ABABABABA ACERTO ABABABABA"
390 PLAY"O4F#DBAEGF#EF#R4"
400 PLAY"O4F#DBAEGF#ED"
410 FOR I=1 TO 2000:NEXT
420 CLS
430 GOTO 440
440 COLOR,5
450 SCREEN 2
460 PSET (32,170)
470 DRAW"F6E6R6E6U6E6U6E6R6F24D6G6R6E6U6R6H24E6F12R6F18D6F6R6U12R6H6U6R12F6E6U12E12R6F6E12D
12R6F18D6G6L30H6L6G6H6G6D12F12R18F6E6R30E
    6R6U18F6R6"
480 DRAW"U138H6L6H18L24G6D6G6D6G6D6G6L6U6H6R12U6H6L6H6L6U6L18G36L6G12D18R6E6F6D12R6E6U12E6H6E12R
    6U6E6R6D6G12D6F6R12D6L6D6L6D12L6G6L1
    2G6L6H6"
490 DRAW"U12G6D12L12G6L6G12L12D6F6D12L18D24R6"
500 DRAW"BM32,170U18L6"
510 DRAW"BM62,146L18"
520 DRAW"BM74,140U6E6NR18U6NH12E6NH12NU12F6NG6R18F6NG6R12E12NU6L18H6NH6G6L6NG6H6U6"
530 DRAW"BM92,134E6F6R6F6R12F6NR6E6R24BE6U12H6G6L24F6G6L6G6"
540 DRAW"BM116,152R6F6ND6E12R6F6"
550 DRAW"BM224,158R6F6R6"
560 DRAW"BM140,110E6H6L6H6BG6BL12D6"
570 DRAW"BM146,68E12U6H6U18E6BL6L12ND18G6L6G12D6G12D6L6D18"
580 DRAW"BM86,182R6D6L6U6"
590 DRAW"BM38,110E6R18U6E6H12U18H6G6D6F6G6D6G6F6G6D6"
600 DRAW"BM26,92E6U6E6H6L6G6D6G6D6R12"
610 DRAW"BM26,26E12U6H6L12D6H6L6D6R6D12F6R6"
620 DRAW"BM86,98R6"
630 FOR I=1 TO 200:NEXT
640 ON X GOTO 650,710,770,830,890,1010,1080,1140,1210,1270,1460,1530,1590,1650,1710,1780,1840,1910,1980
2040,2100,2160,2220,2280,235
0,2410,2470,2540
650 LINE (44,20)-(84,50),13,BF
660 FOR J=1 TO 50:FOR T=1 TO 3
670 LINE (124,155)-(127,158),T,BF
680 FOR H=1 TO 30:NEXT H
690 NEXT T,J
700 GOTO 210
710 LINE (44,20)-(84,30),8,BF:LINE(44,30)-(84,40),15,BF:LINE(44,40)-(84,50),8,BF
720 FOR J=1 TO 50:FOR T=1 TO 3
730 LINE(109,125)-(112,128),T,BF
740 FOR H=1 TO 30:NEXT H
750 NEXT T,J
760 GOTO 210
770 LINE(42,20)-(56,50),1,BF:LINE(56,20)-(70,50),10,BF:LINE(70,20)-(84,50),8,BF
780 FOR J=1 TO 50:FOR T=1 TO 3
790 LINE(77,111)-(79,114),T,BF
800 FOR H=1 TO 30:NEXT H
810 NEXT T,J
820 GOTO 210
830 LINE(44,20)-(84,30),15,BF:LINE(44,30)-(84,40),12,BF:LINE(44,40)-(84,50),8,BF
840 FOR J=1 TO 50:FOR T=1 TO 3
850 LINE (149,144)-(152,147),T,BF
860 FOR H=1 TO 30:NEXT H
870 NEXT T,J
880 GOTO 210
890 A$="R42D18L24H18"
900 DRAW"C15BM44,15"+A$
910 PAINT(46,16),15
920 A$="L42E18R24D18"
930 DRAW"C8BM86,51"+A$
940 PAINT(85,49),8
950 DRAW"BM44,15C15D36"
960 FOR J=1 TO 50:FOR T=1 TO 3
970 LINE(113,117)-(116,120),T,BF
980 FOR H=1 TO 30:NEXT H
990 NEXT T,J
1000 GOTO 210
1010 LINE(44,20)-(51,33),8,BF:LINE(54,20)-(84,33),8,BF:LINE(44,36)-(51,49),8,BF:LINE(54,36)-(84,49),8,BF
1020 LINE(44,33)-(84,36),15,BF:LINE(51,20)-(53,49),15,BF
1030 FOR J=1 TO 50:FOR T=1 TO 3
1040 LINE(90,95)-(93,98),T,BF
1050 FOR H=1 TO 30 :NEXT H
1060 NEXT T,J
1070 GOTO 210
1080 LINE(44,20)-(84,29),8,BF:LINE(44,29)-(84,41),10,BF:LINE(44,41)-(84,50),8,BF
1090 FOR J=1 TO 50:FOR T=1 TO 3
1100 LINE(42,156)-(43,159),T,BF
1110 FOR H=1 TO 30:NEXT H
1120 NEXT T,J
1130 GOTO 210
1140 LINE(44,20)-(51,33),15,BF:LINE(54,20)-(84,33),15,BF:LINE(44,36)-(51,49),15,BF:LINE(54,36)-(84,49),15,B
1150 LINE(44,33)-(84,36),4,BF:LINE(51,20)-(54,49),4,BF
1160 FOR J=1 TO 50:FOR T=1 TO 3
1170 LINE(137,65)-(140,68),T,BF
1180 FOR H=1 TO 30:NEXT H
1190 NEXT T,J
1200 GOTO 210
1210 LINE(42,20)-(56,50),4,BF:LINE(56,20)-(70,50),15,BF:LINE(70,20)-(84,50),8,BF
1220 FOR J=1 TO 50:FOR T=1 TO 3
1230 LINE (57,125)-(60,128),T,BF
1240 FOR H=1 TO 30:NEXT H
1250 NEXT T,J
1260 GOTO 210
```

```
1270 LINE(50,2)-(92,44),4,BF:LINE(50,20)-(92,26),8,BF:LINE(68,2)-(74,44),8,BF
1280 A$="R3F14D3L3H14U3"
1290 DRAW"C8BM50,2"+A$
1300 PAINT(62,14),8
1310 A$="R3D3G14L3U3E14"
1320 DRAW"C8BM89,2"+A$
1330 PAINT(80,14),8
1340 A$="D3R3E14U3L3G14"
1350 DRAW"C8BM50,41"+A$
1360 PAINT(62,32),8
1370 A$="D3L3H14U3R3F14"
1380 DRAW"C8BM92,41"+A$
1390 PAINT(80,32),8
1400 DRAW"C8BM92,41"+A$
1410 FOR J=1 TO 50:FOR T=1 TO 3
1420 LINE(60,98)-(63,101),T,BF
1430 FOR H=1 TO 30:NEXT H
1440 NEXT T,J
1450 GOTO 210
1460 LINE(44,20)-(84,49),7,BF:LINE(49,20)-(52,36),15,BF:LINE(44,27)-(60,30),15,
     BF:LINE(60,24)-(84,27),15,BF
1470 LINE(60,30)-(84,33),15,BF:LINE(44,36)-(84,39),15,BF:LINE(44,42)-(84,45),15,BF
1480 FOR J=1 TO 50:FOR T=1 TO 3
1490 LINE(137,173)-(140,176),T,BF
1500 FOR H=1 TO 30:NEXT H
1510 NEXT T,J
1520 GOTO 210
1530 LINE(44,20)-(84,30),8,BF:LINE(44,30)-(84,40),15,BF:LINE(44,40)-(84,50),4,BF
1540 FOR J=1 TO 50:FOR T=1 TO 3
1550 LINE(80,104)-(83,107),T,BF
1560 FOR H=1 TO 30:NEXT H
1570 NEXT T,J
1580 GOTO 210
1590 LINE(44,20)-(84,30),8,BF:LINE(44,30)-(84,40),15,BF:LINE(44,40)-(84,50),12,BF
1600 FOR J=1 TO 50:FOR T=1 TO 3
1610 LINE(125,131)-(128,134),T,BF
1620 FOR H=1 TO 30:NEXT H
1630 NEXT T,J
1640 GOTO 210
1650 LINE(42,20)-(56,50),12,BF:LINE(56,20)-(70,50),15,BF:LINE(70,20)-(84,50),9,BF
1660 FOR J=1 TO 50:FOR T=1 TO 3
1670 LINE(31,80)-(34,83),T,BF
1680 FOR H=1 TO 30:NEXT H
1690 NEXT T,J
1700 GOTO 210
1710 LINE(44,20)-(51,33),4,BF:LINE(54,20)-(84,33),4,BF:LINE(44,36)-(51,49),4,BF
1720 LINE(54,36)-(84,49),4,BF:LINE(44,33)-(84,36),8,BF:LINE(51,20)-(53,49),8,BF
1730 FOR J=1 TO 50:FOR T=1 TO 3
1740 LINE(18,18)-(21,21),T,BF
1750 FOR H=1 TO 30:NEXT H
1760 NEXT T,J
1770 GOTO 210
1780 LINE(42,20)-(56,50),12,BF:LINE(56,20)-(70,50),15,BF:LINE(70,20)-(84,50),8,BF
1790 FOR J=1 TO 50:FOR T=1 TO 3
1800 LINE(91,154)-(94,157),T,BF
1810 FOR H=1 TO 30:NEXT H
1820 NEXT T,J
1830 GOTO 210
1840 LINE(42,20)-(62,50),15,BF:LINE(64,20)-(84,50),8,BF
1850 DRAW"BM46,26C8R3U3R2D3R3D2L3D3L2U3L3U2"
1860 FOR J=1 TO 50:FOR T=1 TO 3
1870 LINE(89,185)-(92,188),T,BF
1880 FOR H=1 TO 30:NEXT H
1890 NEXT T,J
1900 GOTO 210
1910 LINE(44,20)-(51,33),8,BF:LINE(54,20)-(84,33),8,BF:LINE(44,36)-(51,49),8,BF
1920 LINE(54,36)-(84,49),8,BF:LINE(44,33)-(84,36),4,BF:LINE(51,20)-(54,49),4,BF
1930 FOR J=1 TO 50:FOR T=1 TO 3
1940 LINE(92,71)-(95,74),T,BF
1950 FOR H=1 TO 30:NEXT H
1960 NEXT T,J
1970 GOTO 210
1980 LINE(44,20)-(84,35),15,BF:LINE(44,35)-(84,50),8,BF
1990 FOR J=1 TO 50:FOR T=1 TO 3
2000 LINE(134,104)-(137,107),T,BF
2010 FOR H=1 TO 30:NEXT H
2020 NEXT T,J
2030 GOTO 210
2040 LINE (44,20)-(58,50),12,BF:LINE(58,20)-(84,50),8,BF
2050 FOR J=1 TO 50:FOR T=1 TO 3
2060 LINE(27,163)-(30,166),T,BF
2070 FOR H=1 TO 30:NEXT H
2080 NEXT T,J
2090 GOTO 210
2100 LINE(44,20)-(84,30),1,BF:LINE(44,30)-(84,40),8,BF:LINE(44,40)-(84,50),10,BF
2110 FOR J=1 TO 50:FOR T=1 TO 3
2120 LINE(107,107)-(110,110),T,BF
2130 FOR H=1 TO 30:NEXT H
2140 NEXT T,J
2150 GOTO 210
2160 LINE(44,20)-(84,30),1,BF:LINE(44,30)-(84,40),8,BF:LINE(44,40)-(84,50),10,BF
2170 FOR J=1 TO 50:FOR T=1 TO 3
2180 LINE(92,110)-(95,113),T,BF
2190 FOR H=1 TO 30:NEXT H
2200 NEXT T,J
2210 GOTO 210
2220 LINE(42,20)-(56,50),4,BF:LINE(56,20)-(70,50),10,BF:LINE(70,20)-(84,50),8,BF
2230 FOR J=1 TO 50:FOR T=1 TO 3
2240 LINE(155,137)-(158,140),T,BF
2250 FOR H=1 TO 30:NEXT H
2260 NEXT T,J
2270 GOTO 210
2280 LINE(44,20)-(51,33),7,BF:LINE(54,20)-(84,33),7,BF:LINE(44,36)-(51,49),7,BF:LINE(54,36)-(84,49),7,BF
2290 LINE(44,33)-(84,36),10,BF:LINE(51,20)-(54,49),10,BF
2300 FOR J=1 TO 50:FOR T=1 TO 3
2310 LINE(108,72)-(111,75),T,BF
2320 FOR H=1 TO 30:NEXT H
2330 NEXT T,J
2340 GOTO 210
2350 LINE(44,20)-(84,50),8,BF:LINE(62,25)-(66,45),15,BF:LINE(54,33)-(74,37),15,BF
2360 FOR J=1 TO 50:FOR T=1 TO 3
2370 LINE(86,122)-(89,125),T,BF
2380 FOR H=1 TO 30:NEXT H
2390 NEXT T,J
2400 GOTO 210
2410 LINE(44,20)-(84,50),8,BF:CIRCLE(58,35),8,15,90*3.14/180,270*3.14/180
2420 FOR J=1 TO 50:FOR T=1 TO 3
2430 LINE(185,172)-(188,175),T,BF
2440 FOR H=1 TO 30:NEXT H
2450 NEXT T,J
2460 GOTO 210
2470 LINE(44,20)-(84,50),8,BF:CIRCLE(51,28),4,1,270*3.14/180,90*3.14/180
2480 DRAW"BM56,32C1H5NG1NE2":DRAW"BM49,31C1G1"
2490 FOR J=1 TO 50:FOR T=1 TO 3
2500 LINE(186,77)-(189,80),T,BF
2510 FOR H=1 TO 30:NEXT H
2520 NEXT T,J
2530 GOTO 210
2540 LINE(44,20)-(84,30),4,BF:LINE(44,30)-(84,40),15,BF:LINE(44,40)-(84,50),8,BF
2550 FOR J=1 TO 50:FOR T=1 TO 3
2560 LINE(119,143)-(122,146),T,BF
2570 FOR H=1 TO 30:NEXT H
2580 NEXT T,J
2590 GOTO 210
```

Talent MSX

DISTRIBUIDORES OFICIALES

VIDEO BYTE S.R.L. H. Yrigoyen 40 (1640) Martínez Te.: 793-0597	**COMPUTACION LANUS** Caaguazú 2186 (1824) Lanús Este Te.: 247-0678
MANIAC COMP. Av. Rivadavia 13734 (1704) Ramos Mejía Te.: 654-6844	**ARGESIS COMP. S.A.** Meeks 269 (1832) Lomas de Zamora Te.: 243-1742
FAST Catamarca 1755 (7600) Mar del Plata Te.: (023) 43-430	**MICROSTAR S.A.** Eduardo Costa 892 (1640) Acassuso Te.: 45-0964
CERO-UNO INFORMATICA S.A. Calle 48 Nº 529 (1900) La Plata Te.: (021) 24-9905/9906 24-9907	**MINICOMP S.R.L.** Maipú 862 (2000) Rosario Te.: (041) 64-447 63-091 21-1266

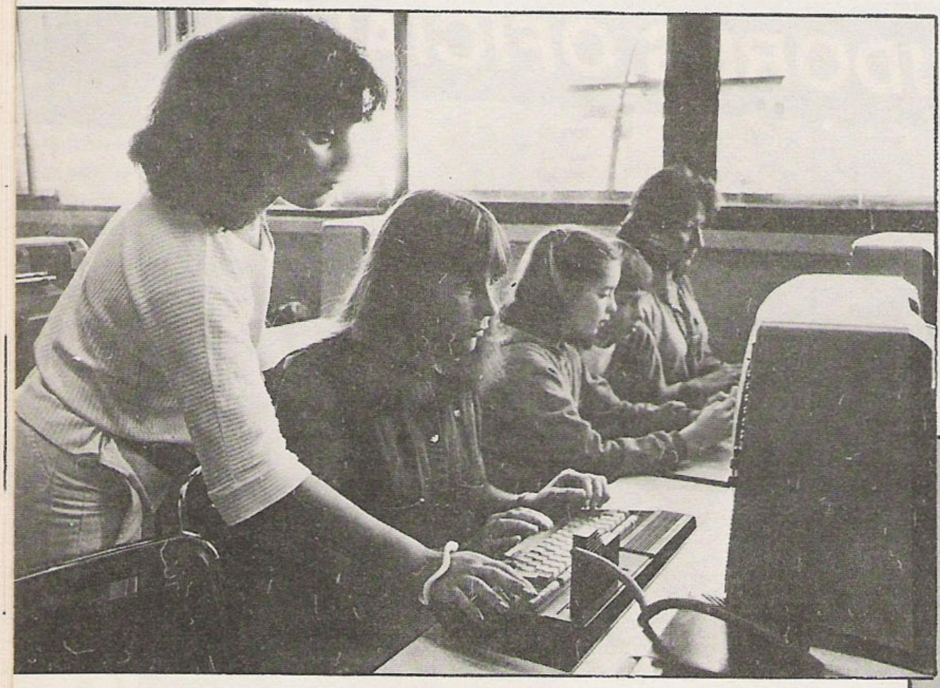


CURSO GRATUITO

El Club de Usuarios MSX anuncia que todo socio o comprador de una computadora Talent MSX tiene derecho a un curso gratuito sobre Introducción a la computación a dictarse en sus idstalaciones.

BANCO DE PROGRAMAS

Ofrecemos un detalle del Banco de Programas que posee el Club, el cual podrá ser utilizado por los socios (siempre haciéndolo en las instalaciones del "Centro para el desarrollo de la Inteligencia").
Obviamente las personas interesadas en el uso de los mismos tendrán todo el asesoramiento del personal, el cual posee un amplio conocimiento en el manejo de ese software.





Software disponible para uso interno en el Club Banco de Programas

GESTION
- *MSX - Plan (Cartridge)*
- *Paso - Calc (Diskette)*

UTILITARIOS
LENGUAJES
ASSEMBLER
- *CMD - ASM (diskette)*
- *MSX - DUAD (diskette)*
- *SIMPLE ASSEMBLER (diskette - cassette)*
- *ZEM - ASSEMBLER (diskette - cassette)*

CMD - CMP (cartridge)
COMPILADOR BASIC (diskette)
FORTH (diskette)
FORTRAN 80 (diskette)
MSX - C (diskette)
RM - COBOL (diskette)
SUPER SOFT - C 80 (diskette)
SUPER SORT (diskette)
UPLISP (diskette)

EDITORES DE TEXTOS
JWJ (WORD PRS) (cartridge -(diskette)
MS - TEXT (diskette - cassette)
TASWORD (diskette)

BASE DE DATOS
D. BASE II (diskette)
MS - BASE (diskette)

EDITORES DE MUSICA
PSG MUSIC EDITOR (diskette)
HAL (diskette)
SUPER SINTETIZER (diskette)

EDITORES DE GRAFICOS
YAMAHA GRAFIC EDITOR (diskette)

EDUCACIONALES
LOGO (cartridge)

Software disponible para la compra presentación en cassettes

GESTION
CONTABILIDAD
FACTURACION
CONTROL DE STOCK
FICHERO
B.G. CALC

UTILITARIOS
ENSAMBLADOR
DESENSAMBLADOR
PASCAL

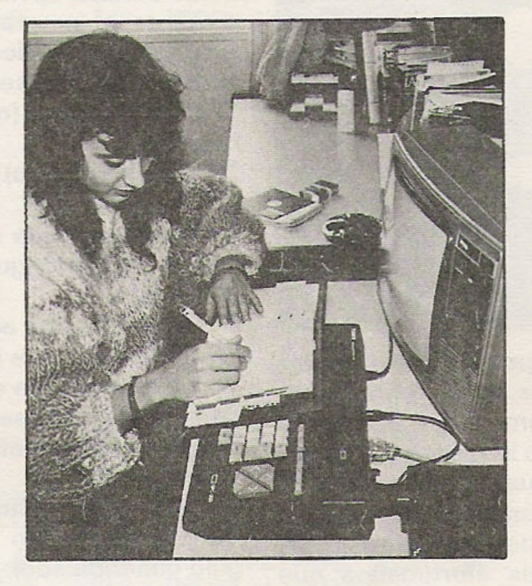
EDUCACIONALES
EUROPA
AFRICA
AMERICA
ASIA
FISICA I
COSMOS
INGLES I
INGLES II

PRESENTACION EN CARTRIDGE
MSX-PLAN
LOGO



AGENDA

Clase: comercial



Este programa, preparado por el Club de usuarios MSX (Avenida Cabildo 2027 Capital), se encargará de mantener un archivo actualizado de nuestra agenda telefónica.
Es muy simple y fácil de usar. Permite que el archivo se encuentre tanto en disco como en cassette.
Cada registro de éste, está dividido en los siguientes campos:
Apellido, Nombre, Dirección, Localidad, Código Postal y Teléfono.
Estos se encuentran perfectamente validados (hablando como programadores expertos).

Esto es: cuando entramos los datos, no hay posibilidad de que nos permita ingresar uno numérico en donde debe entrarse uno alfanumérico.
El menú principal nos ofrece las siguientes opciones:

1- ALTAS
2- BAJAS
3- MODIFICACIONES
4- LISTADO
5- GRABAR
6- CARGAR

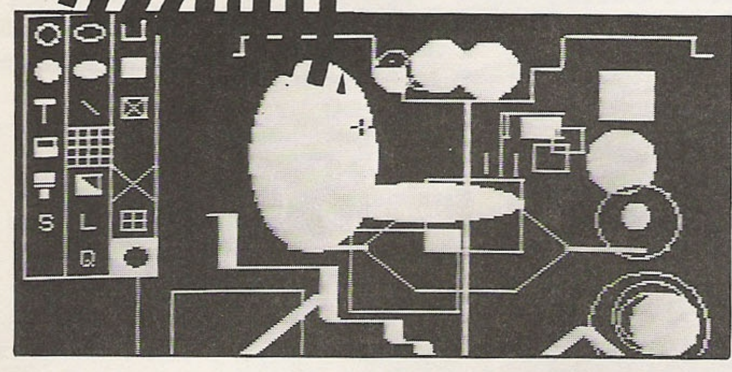
Para mantener este archivo en cassette, habrá que efectuar algunas modificaciones en las instrucciones OPEN, pues el listado que publicamos está preparado para trabajar con la unidad de discos.

Variables importantes

AR$: Matriz alfanumérica que contiene todos los registros.
MA: Valor del último registro entrado.
A$: Contiene los posibles valores numéricos de las opciones y/o respuestas S/N.

```
10 CLEAR 18000:MAXFILES=2:GOSUB
430 ' INICIALIZAR
20 MA=0
30 DIM AR$(500)
40 RESTORE 830:GOSUB 440 :LOCATE
 10,0:PRINT"REGISTROS:";MA ' GENE
RA MENU
50 A$="123456":X=22:Y=20:GOSUB 4
50 'ACEPTA OPCION
60 ON VAL(B$) GOSUB 70,120,190,2
70,390,410:GOTO 40
70 RESTORE 910:GOSUB 440
80 GOSUB 470:D$=D$+SPACE$(87-LEN
(D$))
90 A$="SNsn":X=21:Y=18:GOSUB 450
:IF B$="N" OR B$="n" THEN GOSUB 6
90:GOTO 90
100 GOSUB 540:GOSUB 580
110 A$="SNsn":X=21:Y=21:GOSUB 45
0:IF B$="N" OR B$="n" THEN RETURN
 ELSE 70
120 CLS:LOCATE 0,12,0:PRINT"CUAL
 BORRA?:":X=12:Y=12:A$="":L=25:GO
SUB 590
130 FOR A=25 TO 1 STEP -1:IF MID
$(C$,A,1)<>" " THEN Q$=MID$(C$,1,
A) ELSE NEXT
140 FOR W=1 TO MA:IF INSTR(1,AR$
(W),Q$)=0 THEN NEXT:GOTO 180
150 D$=AR$(W):RESTORE1000:GOSUB
440:F=1:LOCATE 10,1:PRINT"      BA
JAS       ":GOSUB 760
160 A$="SNsn":X=21:Y=18:GOSUB 45
0:IF B$="N" OR B$="n" THEN NEXT:G
OTO 180
170 FOR A=W TO MA-1:AR$(A)=AR$(A
+1):NEXT:MA=MA-1:RETURN
180 CLS:BEEP:LOCATE 0,12,0:BEEP:
PRINT"NO HAY MAS":BEEP:FOR A=1 TO
 20:BEEP:NEXT:RETURN
190 CLS:LOCATE 0,12,0:PRINT"QUE
BUSCA?:":X=11:Y=12:A$="":L=25:GOS
UB 590
200 FOR A=25 TO 1 STEP -1:IF MID
$(C$,A,1)<>" " THEN Q$=MID$(C$,1,
A) ELSE NEXT
210 FOR W=1 TO MA:IF INSTR(1,AR$
(W),Q$)=0 THEN NEXT:GOTO 260
220 D$=AR$(W):RESTORE1000:GOSUB
440:F=1:GOSUB 760
230 A$="SNsn":X=21:Y=18:GOSUB 45
0:IF B$="N" OR B$="n" THEN NEXT:G
OTO 260
240 GOSUB 690:AR$(W)=D$
250 A$="SNsn":X=21:Y=21:GOSUB 45
0:IF B$="N" OR B$="n" THEN 240 EL
SE RETURN
260 CLS:BEEP:LOCATE 0,12,0:BEEP:
PRINT"NO HAY MAS":BEEP:FOR A=1 TO
 20:BEEP:NEXT:RETURN
270 CLS:PRINT"TIENE IMPRESORA(S/
N):":X=21:Y=0:A$="SNsn":GOSUB 450
:IF B$="s" OR B$="S" THEN A=1 ELS
E A=2
280 OPEN "LPT:" FOR OUTPUT AS #1
290 OPEN "CRT:" FOR OUTPUT AS #2
300 CLS:PRINT #A,CHR$(15);TAB(50
)+"LISTADO GENERAL":PRINT #A,"":P
RINT #A,"APELLIDO          ! NOMBRE
                ! DIRECCION
          ! LOCALIDAD          ! C.P
. ! TELEFONO !":FOR Q=1 TO MA:D$=
AR$(Q):LOCATE 0,7,0
310 PRINT #A,MID$(D$,1,15)+" ! "
;
320 PRINT #A,MID$(D$,16,20)+" !
";
330 PRINT #A,MID$(D$,36,25)+" !
";
340 PRINT #A,MID$(D$,61,15)+" !
";
350 PRINT #A,MID$(D$,76,4)+" ! "
;
360 PRINT #A,MID$(D$,80,8)+" ! "
370 FOR W=1 TO 900:NEXT:NEXT:CLO
SE:RETURN
380  RETURN
390 OPEN "DATOS" FOR OUTPUT AS #
1
400 PRINT #1,MA:FOR A=1 TO MA:PR
INT #1,AR$(A):NEXT:CLOSE:RETURN
410 OPEN "DATOS" FOR INPUT AS #1
420 INPUT #1,MA:FOR A=1 TO MA:IN
PUT #1,AR$(A):NEXT :CLOSE:RETURN
430 WIDTH 40:KEY OFF:RETURN
440 CLS:READ B:FOR A=1 TO B:READ
 X,Y,ZZ$:LOCATE X,Y,0:PRINTZZ$:NE
XT:RETURN
450 LOCATE X,Y,1
460 B$=INPUT$(1):IF INSTR(1,A$,B
$)=0 THEN 450 ELSE PRINTB$:LOCATE
 ,,0:RETURN
470 A$="":X=11:Y=5:L=15:GOSUB 59
0:D$=C$
480 A$="":X=11:Y=7:L=20:GOSUB 59
0:D$=D$+C$
490 A$="":X=11:Y=9:L=25:GOSUB 59
0:D$=D$+C$
500 A$="":X=11:Y=11:L=15:GOSUB 5
90:D$=D$+C$
510 A$="1234567890":X=11:Y=13:L=
4:GOSUB 590:D$=D$+C$
520 A$="-1234567890":X=11:Y=15:L
=8:GOSUB 590:D$=D$+C$+SPACE$(8-LE
N(C$))
530 RETURN
540 IF MA=0 THEN MA=1:PO=1:RETUR
N
550 MA=MA+1:FOR A=1 TO MA-1
560 IF AR$(A)>D$ THEN PO=A:RETUR
N ELSE NEXT
570 PO=MA:RETURN
580 FOR A=MA TO PO STEP-1:AR$(A+
1)=AR$(A):NEXT:AR$(PO)=D$:RETURN
590 C$=STRING$(L,32)
600 A=1
610 LOCATE X,Y,0:PRINTC$;:LOCATE
 X+A-1,Y,1:B$=INPUT$(1)
620 IF ((A$="" AND ASC(B$)>31 AN
D ASC(B$)<>127) OR (INSTR(1,A$,B$
)<>0)) AND A<=L THEN LOCATE X,Y,0
:MID$(C$,A,1)=B$:A=A+1
630 IF ASC(B$)=28 AND A<L+1 THEN
 A=A+1
640 IF ASC(B$)=127 AND A<L+1 THE
N C$=MID$(C$,1,A-1)+MID$(C$,A+1,L
-A)+" "
650 IF ASC(B$)=29 AND A>1 THEN A
=A-1
660 IF ASC(B$)=8 AND A>1 THEN C$
=MID$(C$,1,A-2)+MID$(C$,A,L-A+1)+
" ":A=A-1
670 IF ASC(B$)<>13 THEN 610
680 Q=INSTR(1,C$," "):IF Q=0 THE
N LOCATE X,Y,0:PRINTC$:RETURN ELS
E MID$(C$,Q,1)=" ":GOTO 680
690 A$="":X=11:Y=5:L=15:C$=MID$(
D$,1,15):GOSUB 600:MID$(D$,1,15)=
C$
700 A$="":X=11:Y=7:L=20:C$=MID$(
D$,16,20):GOSUB 600:MID$(D$,16,20
)=C$
710 A$="":X=11:Y=9:L=25:C$=MID$(
D$,36,25):GOSUB 600:MID$(D$,36,25
)=C$
720 A$="":X=11:Y=11:L=15:C$=MID$
(D$,61,15):GOSUB 600:MID$(D$,61,1
5)=C$
730 A$="1234567890":X=11:Y=13:L=
4:C$=MID$(D$,76,4):GOSUB 600:MID$
(D$,76,4)=C$
740 A$="-1234567890":X=11:Y=15:L
=8:C$=MID$(D$,80,8):GOSUB 600:MID
$(D$,80,8)=C$
750 RETURN
760 LOCATE 11,5,0:PRINTMID$(D$,1
,15)
770 LOCATE 11,7:PRINTMID$(D$,16,
20)
780 LOCATE 11,9:PRINTMID$(D$,36,
25)
790 LOCATE 11,11:PRINTMID$(D$,61
,15)
800 LOCATE 11,13:PRINTMID$(D$,76
,4)
810 LOCATE 11,15:PRINTMID$(D$,80
,8)
820 RETURN
830 DATA 8,10,3,"MENU PRINCIPAL"
840 DATA 5,6,"<1> ALTAS"
850 DATA 5,8,"<2> BAJAS"
860 DATA 5,10,"<3> MODIFICACIONE
S"
870 DATA 5,12,"<4> LISTADO"
880 DATA 5,14,"<5> GRABAR"
890 DATA 5,16,"<6> CARGAR"
900 DATA 9,20,"OPCION(1-6)?:"
910 DATA 9,15,1,"ALTAS"
920 DATA 0,5,"APELLIDO :
       :"
930 DATA 0,7,"NOMBRE    :
              :"
940 DATA 0,9,"DIRECCION:
                  :"
950 DATA 0,11,"LOCALIDAD:
         :"
960 DATA 0,13,"COD. POS.:      :"
970 DATA 0,15,"TELEFONO :
  :"
980 DATA 5,18,"TODO BIEN(S/N)?:"
990 DATA 5,21,"MAS ALTAS(S/N)?:"
1000 DATA 9,10,1,"MODIFICACIONES
"
1010 DATA 0,5,"APELLIDO :
        :"
1020 DATA 0,7,"NOMBRE    :
              :"
1030 DATA 0,9,"DIRECCION:
                  :"
1040 DATA 0,11,"LOCALIDAD:
         :"
1050 DATA 0,13,"COD. POS.:      :
"
1060 DATA 0,15,"TELEFONO :
   :"
1070 DATA 5,18,"ES ESTE  (S/N)?:
"
1080 DATA 5,21,"TODO BIEN(S/N)?:
"
1090 DATA 0,7,"NOMBRE   :"
1100 DATA 0,9,"DIRECCION:"
1110 DATA 0,11,"LOCALIDAD:"
1120 DATA 0,13,"COD. POS.:"
1130 DATA 0,15,"TELEFONO :"
```

DISENADOR DE PANTALLAS



Con este programa se grafican circunferencias, cuadrados y líneas de diversos tamaños. También se pueden trazar líneas y curvas como si estuviéramos dibujando con un lápiz, permitiéndonos delinear figuras con mayor precisión y a nuestro antojo.

Aparte de las ventajas para diseñar, contamos con efectos que nos pueden ayudar a dar un toque más atractivo a los bosquejos como, por ejemplo, cuadricular la pantalla, invertir los colores o rellenar figuras irregulares.

Aclaramos irregulares porque además de la opción para hacer circunsferencias, contamos con la posibilidad de graficar círculos, rectángulos y elipces rellenas. También existe la opción que permite dibujar como si estuviéramos pintando con una fibra con punta chata. Este tipo de marcadores da una sensación de sombra a los trazos.

Pero a pesar de todas estas ventajas, este programa sería incompleto si no tuviera incluido los recursos de cargar y guardar en cassettes, como también el permitir la salida por impresora MSX (para los afortunados que tengan este periférico). La posición de nuestro "lápiz" está marcada por un sprite con forma de cruz.
El manejo de este programa es sencillo y tiene un menú que nos explica algunos puntos. Por ejemplo, en el ángulo superior izquierdo, aparecerán graficados todos los símbolos correspondientes a los recursos posibles. Para seleccionar algún tipo de recurso, posicionemos el cursor (cruz) sobre la opción y luego presionemos la barra espaciadora. El símbolo correspondiente a esa opción, se dibujará en un casillero blanco.
Otra posibilidad que contamos es el "ESPEJO", que permite dibujar efectos calidoscópicos.

Estas son las opciones que presenta el programa. Entre paréntesis indica el símbolo que lo representa.

1- Circunferencia (circunferencia).
2- Círculo (círculo).
3- Línea (regla T).
4- Pintura —pinta áreas de figuras— (rectángulo semi pintado).
5- Pincel (brocha ancha).
6- Grabar pantalla (S).
7- Elipse —dibuja este tipo de figuras— (elipse).
8- Elipse rellena (elipse con el área marcada).
9- Lápiz (línea oblicua).
10- Cuadriculado (rectángulo cuadriculado).
11- Inverso —invierte el color de fondo con el de tinta— (rectángulo pintado de la diagonal para abajo).
12- Carga pantalla (L).
13- Para abandonar el programa (Q).
14- Cuadrado (cuadrado).
15- Cuadrado relleno (cuadrado con el área pintada).
16- Borra la pantalla para volver volver a graficar (rectángulo con dos diagonales marcadas).
17- Imprime por impresora (espacio en blanco).
18- Espejo (rectángulo con las diagonales marcadas).

VARIABLES IMPORTANTES

X: Posición horizontal del sprite.
Y: Posición vertical del sprite.
TBL: Matriz con código de opciones.
SPRITE 0: Cruz.

ESTRUCTURA DEL PROGRAMA

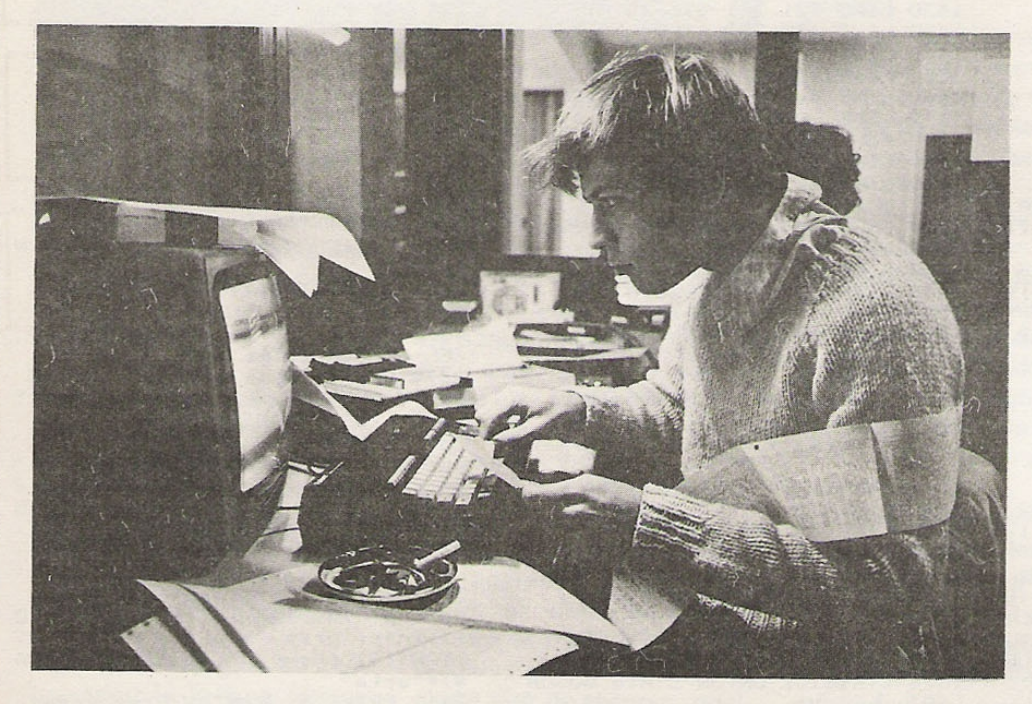
20-50: Pinta como pincel.
60-180: Lee la opción que elegimos.
190-230: Visualiza la imagen de la opción elegida.
260-310: Fin de proceso de elección, rutina de diseño, dibuja círculos, define centro y coordenadas.
330-350: Pregunta si el círculo se debe pintar.
370-460: Dibuja elipse.
480-490: Pinta elipse.
510-520: Dibuja línea.
540-550: Pinta área de una figura cerrada.
630-690: Cuadrícula.
710-780: Invierte colores de fondo con el de tinta.
800: Pincel.
820-840: Dibuja cuadrados.
870-880: Limpia pantalla.
900-960: Graba pantalla.
980-1060: Carga pantalla.
1080-1190: Rutina de impresión en papel.
1210-1220: Abandona el programa.
1240-1310: Inicio del programa.
1340-1720: Dibuja los símbolos de las opciones en una tabla menú.
1750-1870: Loop que mueve el cursor.
1890-2290: Inicializa variables y sprites.
2300-2610: Presenta explicaciones en pantalla.

```
10 GOTO 1240
20 PS=5
30 IF X+PS>255 THEN PS=PS-1
40 LINE (X+4,Y+3)-STEP(PS,0):RE
TURN
50 PSET(X+3,Y+4):PSET(X+3,195-Y
):PSET(256-(X-44),Y+4):PSET(256-
(X-44),195-Y):RETURN
60 PLAY "L2405C06C"
70 IF X+3>47 THEN GOTO 210
80 IF Y>112 THEN GOSUB 2640:RET
URN
90 V=INT((X+3)/16):W=INT((Y+4)/
16)
100 H=TBL(V,W)
110 IF H=6 THEN GOSUB 900:GOSUB
 2640:RETURN
120 IF H=7 THEN GOSUB 2640:RETU
RN
130 IF H=11 THEN GOSUB 630:GOSU
B 2640:RETURN
140 IF H=12 THEN GOSUB 710:GOSU
B 2640:RETURN
150 IF H=13 THEN GOSUB 980:GOSU
B 2640:RETURN
160 IF H=14 THEN GOSUB 1200:GOS
UB 2640:RETURN
170 IF H=18 THEN GOSUB 1080:GOS
UB 2640:RETURN
180 IF H=19 THEN GOSUB 870:GOSU
B 2640:RETURN
190 F=H
200 GOSUB 2190:RETURN
210 ON F GOSUB 260,260,510,540,
800,,,370,370,560,,,,,820,820,86
0,,,600
220 GOSUB 2640
230 RETURN
240 '
250 '
260 IF CF=0 THEN CF=1:C1=X+3:C2
=Y+4:PUT SPRITE 1,(C1-1,C2-2),1:
FOR I = 1 TO 8: CSR(I,2)=0:NEXT:
RETURN
270 FOR I=1 TO 8:CSR(I,2)=YG(I)
:NEXT
280 CF=0:PUT SPRITE 1,(0,209),1
290 IF C1-ABS(X+3-C1)<48 THEN P
LAY"L2402CE":RETURN
300 CIRCLE(C1,C2),ABS(X+3-C1),,
,,8/7
310 IF F<2 THEN RETURN
320 '
330 IF POINT(C1,C2)=15 THEN PLA
Y"L2403CE":RETURN
340 PAINT(C1,C2):RETURN
350 RETURN
360 '
370 IF OV=0 THEN OV=1:01=X+3:02
=Y+4:PUT SPRITE 1,(01-1,02-2),1:
FOR I=1 TO 8:CSR(I,2)=0:NEXT:RET
URN
380 '
390 IF OV=1 THEN OV=2:R1=ABS(X+
3-01):PUT SPRITE 2,(X+2,Y+2),1,1
:X=01-3:Y=02-4:FOR I=1 TO 8:CSR(
I,2)=YG(I):CSR(I,1)=0:NEXT:RETUR
N
400 R2=ABS(Y+4-02):RD=R1
410 OV=0
420 IF R1<R2 THEN RD=R2
430 FOR I=1 TO 8:CSR(I,1)=XG(I)
:NEXT
440 PUT SPRITE 1,(0,209):PUT SP
```

```
450 IF ST RD<48 THEN PLAY L2402
CE":RETURN
460 CIRCLE(O1,O2),RD,,,,R2/R1
470 '
480 IF F=8 THEN RETURN
490 PAINT (O1,O2):RETURN
500 '
510 IF LF=0 THEN LF=1:L1=X+3:L2
=Y+4:PUT SPRITE 1,(L1-1,L2-2),1:
RETURN
520 LINE (L1,L2)-(X+3,Y+4):LF=0
:PUT SPRITE 1,(0,209):RETURN
530 '
540 PAINT(X+3,Y+4)
550 RETURN
560 '
570 SWAP PU,PD
580 RETURN
590 '
600 SWAP MU,MD
610 RETURN
620 '
630 FOR I=0 TO 192 STEP 8
640 LINE (48,I)-STEP (255,0)
650 NEXT I
660 FOR I=48 TO 256 STEP 8
670 LINE (I,0)-STEP (0,191)
680 NEXT I
690 RETURN
700 '
710 VDP(7)=FC
720 FOR I=0 TO 6144
730 IF VPEEK (I)=0 AND VPEEK(BA
SE(11)+I)=FC THEN VPOKE BASE(11)
+I,BC:GOTO 760
740 IF VPEEK(I)=0 AND VPEEK(BAS
E(11)+I)=BC THEN VPOKE BASE(11)+
I,FC:GOTO 760
750 VPOKE I,NOT(VPEEK(I)) AND 2
55
760 NEXT
770 SWAP BC,FC
780 RETURN
790 '
800 SWAP BU,BD:RETURN
810 '
820 IF BF=0 THEN BF=1:B1=X+3:B2
=Y+4:PUT SPRITE 1,(B1-1,B2-2),1,
1:RETURN
830 IF F=15 THEN LINE (B1,B2)-(
X+3,Y+4),,B:BF=0:PUT SPRITE 1,(0
,209):RETURN
840 LINE (B1,B2)-(X+3,Y+4),,BF:
BF=0: PUT SPRITE 1,(0,209):RETUR
N
850 '
860 SWAP EU,ED: RETURN
870 LINE(48,0)-(255,191),4,BF
880 RETURN
890 '
900 OPEN "CAS:PIC" FOR OUTPUT A
S#2
910 FOR I=48 TO 5836 STEP 256
920 FOR J=I TO I+207
930 PRINT #2,VPEEK(J);",";VPEEK
(8192+J)
940 NEXT J,I
950 CLOSE #2
960 RETURN
970 '
980 GOSUB 870
990 OPEN "CAS:PIC" FOR INPUT AS
#2
1000 FOR I=48 TO 5836 STEP 256
1010 FOR J=1 TO I+207
1020 INPUT #2, VP,VC
1030 VPOKE J,VP:VPOKE 8192+J,VC
1040 NEXT J,I
1050 CLOSE #2
1060 RETURN
1070 '
1080 LPRINT CHR$(27);"L";CHR$(2
08);CHR$(0)
1090 FOR I=0 TO 183 STEP 8
1110 FOR J=48 TO 255
1120 B$=""
1130 FOR K=0 TO 7
1140 IF POINT(J,J+K)=FC THEN B$
=B$+"1" ELSE B$=B$+"0"
1150 NEXT K
1160 NEXT J,I
1170 LPRINT CHR$(VAL("&B"+B$))
1180 '
1190 RETURN
1200 '
1210 COLOR 15,4,4:SCREEN 0,0
1220 END
1240 DEFINT A-Z:MAXFILES=2
1250 GOSUB 1890
1260 GOSUB 2300
1270 BC=4:FC=15
1280 COLOR FC,BC,BC
1290 SCREEN 2,0,0
1300 OPEN "GRP:"AS#1
1310 GOSUB 2030
1320 '
1330 '
1340 LINE(0,0)-(47,111),,B
1350 FOR I=15 TO 31 STEP 16
1360 LINE(I,0)-(I,111)
1370 NEXT I
1380 CIRCLE(8,8),4:CIRCLE STEP(
15,0),5,,,,1/2:LINE STEP (12,-4)
-STEP(7,7),,B
1390 CIRCLE(8,23),BC:PAINT(8,23
):CIRCLE STEP(15,0),5,,,,1/2:LIN
E STEP(12,-4)-STEP(7,7),,BF:PAIN
T STEP(-15,-2)
1400 LINE (7,36)-STEP(0,8)
1410 LINE (4,35)-STEP(6,1),,BF
1420 LINE (21,36)-STEP(5,5)
1430 LINE (35,35)-STEP(8,8),,B
1440 LINE (35,35)-STEP(8,8)
1450 LINE(35,43)-STEP(8,-8)
1460 LINE(4,51)-STEP(7,4),,B
1470 LINE -STEP(-7,3),,BF
1480 FOR I=15 TO 31 STEP 4
1490 LINE(I,47)-STEP(0,15)
1500 LINE (15,32+I)-STEP(15,0)
1510 NEXT
1520 DRAW"BM35,110R6D6I6U6R6D3R
3D6I6U3"
1530 LINE(4,66)-STEP(6,3),,BF
1540 LINE(4,71)-STEP(6,0)
1550 LINE(6,73)-STEP(2,3),,BF
1560 LINE(19,67)-STEP(8,8),,B
1570 LINE(19,67)-STEP(8,8):PAIN
T STEP(-3,-1)
1580 LINE(32,64)-STEP(15,15)
1590 LINE(47,63)-STEP(-15,15)
1600 LINE(32,95)-STEP(15,15),,B
F
1610 PSET(6,84):PRINT#1,"S"
1620 LINE(35,83)-STEP(8,8),,B
1630 LINE(39,83)-STEP(0,8)
1640 LINE(35,87)-STEP(8,0)
1650 PSET(21,100):PRINT#1,"Q"
1660 PSET(21,84):PRINT#1,"L"
1670 '
1680 X=48:Y=96:PU=-1:PD=0:EU=-1
:ED=0:BU=-1:BD=0:MU=-1:MD=0
1690 STRIG(0) ON:ON STRIG GOSUB
 60
1700 RESTORE 2130
1710 FOR I=1 TO 8:FOR J=1 TO 2
1720 READ A:CSR(I,J)=A:NEXT J,I
1730 '
1740 '
1750 PUT SPRITE 0,(X,Y),1,0
1760 IF PD AND X>47 THEN PSET (
X+3,Y+4)
1770 IF ED AND X>47 THEN PRESET
 (X+3,Y+4)
1780 IF BD AND X>47 THEN GOSUB
20
1790 IF MD AND X>47 THEN GOSUB
50
1800 IF STICK(0)=0 THEN 1800
1810 PUT SPRITE 0,(X,Y),1,0
1820 X=X+CSR(STICK(0),1):Y=Y+CS
R(STICK(0),2):
1830 IF X<-3 THEN X=-3
1840 IF X>252 THEN X=252
1850 IF Y>187 THEN Y=187
1860 IF Y<-3 THEN Y=-3
1870 GOTO 1750
1880 '
1890 RESTORE 2130
1900 DIM CSR(8,2)
1910 DIM TBL(3,7)
1920 DIM ER(8,8)
1930 DIM YG(8),XG(8)
1940 FOR I=1 TO 8
1950 FOR J=1 TO 2
1960 READ A:CSR(I,J)=A:IF JMOD2
=0 THEN YG(I)=A ELSE XG(I)=A
1970 NEXT J,I
1980 FOR I=0 TO 2
1990 FOR J=0 TO 6
2000 READ TBL(I,J):NEXT J,I
2010 RETURN
2020 '
2030 RESTORE 2160
2040 FOR K=0 TO 1
2060 FOR I=0 TO 7
2070 READ A:S$=S$+CHR$(A)
2080 NEXT
2090 SPRITE$(K)=S$
2100 NEXT K
2110 RETURN
2120 '
2130 DATA 0,-1,1,-1,1,0,1,1,0,1
,-1,1,-1,0,-1,-1
2140 DATA 1,2,3,4,5,6,7,8,9,10,
11,12,13,14,15,16,17,18,19,20,21
2150 '
2160 DATA 16,16,16,195,16,16,16
,0
2170 DATA 64,224,64,0,0,0,0,0
2180 '
2190 IF F=21 THEN RETURN
2200 BF=0:CF=0:OV=0:LF=0
2210 PUT SPRITE 1,(0,209):PUT S
PRITE 2,(0,209)
2220 V1=V*16:W1=W*16
2230 FOR I=1 TO 14
2240 FOR J=1 TO 14
2250 IF POINT(V1+I,W1+J)=BC THE
N C=FC ELSE C=BC
2260 PSET(32+I,96+J),C
2270 NEXT J,I
2280 GOSUB 2640
2290 RETURN
2300 '
2310 '
2320 '
2330 SCREEN 0,0:KEY OFF
2340 PRINT TAB(5);"/// SISTEMA
DE DISE"+CHR$(&HA5)+"IO///":PRIN
T:PRINT
2350 PRINT TAB(1);"SELECCIONE L
A OPCION"
2360 PRINT TAB(1);"POSICIONANDO
 EL CURSOR SOBRE LA"
2370 PRINT TAB(1);"OPCION DEL M
ENU Y PRESIONE LA BARRA":PRINT T
AB(1);"DE ESPACIO":PRINT
2380 PRINT"1. CIRCULO: Pres. es
pacio p/ definir":PRINT TAB(12)"
centro, nuevamente":PRINT TAB(12
)"espacio p/ definir rad.y":PRIN
T TAB(12)"dise"+CHR$(&HA4)+"ar"
2390 PRINT"2. CIRCULO .... Como
 arriba"
2400 PRINT "3. LINEA: Pres. esp
acio p/ iniciar la":PRINT TAB(10
)"linea, pres. espacio de":PRINT
 TAB(10)"nuevo p/ dise"+CHR$(&HA
4)+"arla"
2410 PRINT"4. PINTA: Pres.espac
io p/ pintar el":PRINT TAB(10)"a
rea"
2420 PRINT "5. PINCEL:Pres.espa
cio p/ dise"+CHR$(&HA4)+"ar."
2430 PRINT "6. SAVE(S): Salva e
n cinta."
2440 PRINT:PRINT" Mas..."
2450 R$=INKEY$:IF R$="" THEN 24
50
2460 CLS
2470 PRINT "7. ELIPSE: como CIR
CULO, mas definir":PRINT TAB(11)
;"radio tambien."
2480 PRINT "8. ELIPSE RELLENA:
como arriba."
2490 PRINT"9. LAPIS: Como PINCE
L"
2500 PRINT"10. CUADRICULADO: Di
se"+CHR$(&HA4)+"ar cuadricula"
2510 PRINT"11. REVES: Revierte
color de fondo":PRINT TAB(11)"co
n el de tinta."
2520 PRINT"12. LOAD: Carga grav
acion desde cinta."
2530 PRINT "13. CUADRADO: Como
para linea"
2540 PRINT"14. CUADRADO... Como
 arriba"
2550 PRINT"15. BORRAR: Como par
a LAPIS."
2560 PRINT "16.IMPRIMIR : Saca
por impresora"
2570 PRINT"17. LIMPA PANTALLA"
2580 PRINT"18. ESPEJO: Como par
a LAPIS"
2590 PRINT :PRINT"Ejecutar..."
2600 R$=INKEY$:IF R$="" THEN 26
00
2610 RETURN
2640 RESTORE 2130
2650 FOR I=1 TO 8:FOR J=1 TO 2
2660 READ A:CSR(I,J)=A:NEXT J,I
2670 RETURN
```

PROGRAMACION APLICADA

La mejor forma de aprender BASIC es ir programándolo a medida que lo necesitemos.



Primero sería conveniente interiorizarnos en el tema sobre el que vamos a programar.

Una vez que tengamos conocimiento de los puntos más importantes del problema que la máquina nos ayudará a resolver, anotemos por orden de importancia cuáles son los enigmas que solucionará el programa.

Propongamos un ejemplo y programémoslo. Supongamos que queremos estudiar los verbos. Entonces primero sería conveniente repasar un poco este tema y rescatar los puntos principales. Sabemos que existen dos clases de verbos: irregulares y regulares. Hagamos un programa para los verbos que pertenecen a este último grupo. Los verbos se conjugan según el modo, tiempo, persona y número.

Los verbos en infinitivo pueden terminar en "ar", "er" o "ir".
Los terminados en "ar" pertenecen a la "primera conjugación", los terminados en "er" a la "segunda conjugación", y los que terminan en "ir" a la "tercera". Lo más importante de tener en cuenta es que los verbos regulares de la primera conjugación, tienen las mismas terminaciones.

Supongamos que nuestros intereses primordiales son:
- Explicación básica sobre los verbos con preguntas sobre lo enseñado.
- Conjugación de un verbo regular.
Al ser estas dos tareas diferentes, habrá que tratarlas por separado.
Realicemos un diagrama (de bloque) para desarrollar un poco más cada una de las opciones que marcamos arriba.
En la figura 1 se muestra el diagrama de bloques para la primera opción; mientras que en la figura 2 aparece el diagrama para la segunda.
Recordemos entonces que los diagramas de bloques muestran en forma general, las operaciones a realizar por el programa.
Comencemos por la primera opción, que abarca el tema de la explicación sobre las propiedades verbales.
En esta parte habrá que entrar en la computadora toda la información que esperamos ver luego a modo de explicación.
Cuando comencemos a programar, tratemos de colocar mensajes aclaratorios en el listado para que cuando los leamos, identifiquemos inmediatamente qué hace la porción de jeroglíficos que leemos. La instrucción REM o el apóstrofe, permite colocar mensajes aclaratorios en el listado sin afectar el programa. Es decir que la computadora no "leerá" nada que haya detrás del REM o del apóstrofe.
Copiemos la línea 10 de la figura 3:
10 REM EXPLICACIONES DE LOS VERBOS.
Habíamos propuesto que salgan por pantalla las explicaciones sobre los verbos. Pero antes de mostrar por pantalla algún mensaje, es conveniente asegurarnos que la pantalla esté limpia para que no se mezcle con nuestros mensajes. Entonces busquemos en el manual de la computadora alguna instrucción que despeje el "pizarrón".
La instrucción CLS limpia la pantalla. Entonces, ésta va a ser la primera instrucción del programa.
En las MSX, en la parte inferior del monitor aparecen escritos los comandos que realizan las teclas de función. Para borrar estas impresiones (porque nada tienen que ver con nuestro programa) usemos KEY OFF.
Otro "chiche" para programar, es escoger un color de fondo y otro de tinta para que los mensajes aparezcan a nuestra vista con algún efecto atractivo.

Con COLOR A,B escogemos como color de tinta a A y de fondo o papel a B.
Podríamos pedir verde oscuro como color de tinta y negro para el papel.
Tipiemos la segunda línea del programa que es la número 20 de la figura 3.

¿Qué nos va a explicar el programa?

Primero nos dirá cuáles son los accidentes verbales y luego irá desarrollando cada uno de ellos. Con PRINT "cadena de caracteres" imprimimos en la pantalla desde el primer renglón superior o debajo del último mensaje Que apareció en la pantalla, todos los caracteres encerrados entre comillas.
Es hora de copiar la línea 30. Aquí, entre los dos puntos ":" y la "M" hay ocho espacios en blanco. Esto se debe a que en la pantalla, de la forma que estamos trabajando, sólo entran 37 caracteres a lo ancho. Desde el primer espacio en blanco después de las comillas, hasta los dos puntos (:) inclusive, hay 29 lugares, entonces con 8 espacios más, habría 38. O sea, pasaríamos a la primera columna del siguiente renglón.
Desarrollemos cada uno de los cuatro accidentes verbales, comenzando por MODO.
PRINT sin nada entre comillas deja un renglón en blanco.
Copiemos la línea 40. Las cuatro comas que aparecen entre las comillas (estas no encierran las comas sino que

pertenecen a las comillas de cierre y apertura de una cadena de caracteres) dejan un renglón en blanco. Esto equivaldría a decir una vez PRINT; pero esto lleva más tiempo que escribir sólo 4 comas.

Una vez aclarado el significado de la primera propiedad, entremos los distintos modos que existen. Podemos entrar ahora la línea 50.

A cada explicación es conveniente que la acompañe un ejemplo.

PRINT TAB(n) "cadena de caracteres", comenzará a imprimir los caracteres entre comillas a partir de la columna n, contando a partir de la izquierda. Tipeemos la línea 60.

La número 70 no trae nada nuevo y es claro que imprime en pantalla... la definición del SUBJUNTIVO, ¡muy bien!! Antes de seguir, contemos cuantos renglones ocupamos en la pantalla. Son en total 18, nos quedan sólo 5 disponibles.

Entonces imprimamos algo más pero reservemos las dos últimas líneas de la pantalla para mensajes aclaratorios. Se pueden imprimir la cantidad de mensajes que querramos, pero no es conveniente superar los 24 renglones (una pantalla) porque si ésto sucede, las líneas subirán escapándose de nuestra visión sin permitirnos entender lo que contenían. Entonces cada vez que hayamos terminado de imprimir la pantalla, imprimiremos un mensaje: "PRESIONE ALGUNA TECLA PARA CONTINUAR" y luego volvamos al asunto que estábamos atendiendo antes del cartel aclaratorio.

La instrucción GOSUB n, hace que el programa al llegar a esta orden, salte hasta la línea n, realice todas las operaciones que hay hasta encontrar una orden RETURN. Cuando un RETURN es hallado, la máquina vuelve a la siguiente instrucción del GOSUB n. Entremos la línea 80. Aquí, se imprime un ejemplo y luego saltamos a la línea lejana, como podría ser la línea 590. En ésta, pedimos que el cursor se ubique en la columna cero (la primera); línea 22 y luego que imprima una aclaración. Pero aún no hemos puesto el RETURN porque queremos que haga algo más. En la línea 600 decimos que

si no hay ninguna tecla presionada en-
IF INKEY$=""
tonces vuelva a la línea 600, sino
THEN ELSE
que limpie la pantalla y luego que
CLS
vuelva a la siguiente instrucción des-
RETURN
de donde se desvió.

Aquí vimos algunas cosas nuevas como la instrucción IF... THEN... ELSE. Esto se lee así: si (IF) pasa tal cosa, entonces (THEN) hacer ésto, sino (ELSE) hacer esto otro.

Los dos puntos ":" separan instrucciones, ésta evita que tengamos que escribir una instrucción por renglón (figura 4).

INKEY$= lee el caracter que se presionó desde el teclado. Las dos comillas seguidas significa que no se tocó ninguna tecla.

Una vez que encontró el RETURN, vuelve, en nuestro caso, a la línea 90 porque es la siguiente instrucción después del GOSUB 590.

Seguimos entrando información y ejemplos sobre los modos verbales y sin nada nuevo hasta la línea 120 inclusive. Al terminar con un ítem, lo mejor es averiguar si se entendió, copiemos la línea 130.

En la 140 comenzamos con una pequeña batería de preguntas. En la línea 150 volvemos a usar el recurso GOSUB pero esta vez nos vamos a la línea 610.

Aquí imprimimos los modos que explicamos y pasamos a la línea siguiente, la 620; donde preguntaremos con un INPUT el número que escogemos como respuesta. La instrucción INPUT "mensaje" ;X (basta con fijarse en el manual de la computadora para mayor información de esta sentencia) acepta un valor para la variable X. Después del INPUT, hay un RETURN, entonces volvemos a la línea 160. En esta línea decimos que si
IF
nuestra respuesta fue 3 entonces
X=3 THEN
pasemos (sin volver) a la línea 180.
GOTO 180

Observemos que la diferencia que existe entre GOTO y GOSUB es que la primera salta hasta una línea determinada pero no vuelve como el GOSUB. Entonces no hace falta usar el GOTO con un RETURN.

También puede suceder que nuestra respuesta no sea 3. En ese caso, la siguiente instrucción después del IF... THEN, en nuestro programa, pasa a la línea 170, donde aclaramos con sonido (BEEP) y un mensaje, que hemos fallado en la contestación. Salta nuevamente a la línea 590. Cuando encuentre un RETURN pasa a la instrucción siguiente al GOSUB, o sea, GOTO 30. Ahora la computadora pasa a la línea 30 que es el principio de la explicación de los verbos.

C
Explicación del primer accidente verbal
Examinar si se entendió el tema explicado
Explicar segundo ítem verbal
Examinar la segunda explicación
Explicar el tercer ítem
Examinar lo explicado en el tercer ítem
Explicar cuarto ítem
Examinar última explicación
F

Figura 1

C
Presentación de pantalla con menú de opciones
Conjuga verbo en indicativo
Conjuga verbo en subjuntivo
Conjugación para el potencial
Conjugación para imperativo
F

Figura 2

En la 180 damos otra pregunta. En la 190 decimos que si elegimos la opción 1, pasamos a la tercera pregunta pero si fallamos, volvemos a comenzar la explicación. Así sigue hasta la línea 270. Acá explicamos otro ítem: el tiempo verbal. En las dos líneas siguientes mostramos ejemplos. A partir de la 300, volvemos a transformarnos en verdugos examinadores porque comenzamos nuevamente a tomar lección. Las respuestas se reducen en contestar SI o NO. Entremos hasta la línea 320. Preguntamos, y si nuestra respuesta no es "SI" o "si", entonces volvemos a comenzar la explicación del tiempo.

Figura 4

```
10 CLS  ┐
20 PRINT┘=10 CLS:PRINT
```

En la 330 hacemos otra pregunta y 340 es la verificación de nuestra respuesta. Sigue igual, sin nada nuevo, hasta la línea 390 donde pedimos limpiar la pantalla y dar la definición de "persona"; otro ítem de los accidentes verbales.

A partir de aquí no existe nada nuevo. Sólo usamos PRINT, IF... THEN, GOSUB y GOTO.

En la línea 400 damos los ejemplos de personas, en la 410 definimos "número'6 para las propiedades verbales. Desde 420, hasta la línea 510, hacemos preguntas.

En la línea 530 hasta la 570 damos algunas definiciones más sobre verbos, pero, sin hacer preguntas. En la 580 aclaramos que se acabó la enseñanza y volvemos al menú. Las líneas 590 hasta 620 ya las comentamos.

¿Qué hace falta decirle a la computadora para que conjugue un verbo?

Vamos a realizar la segunda parte del programa.

¿Qué pretendemos que haga la computadora en esta parte?

Nos tendría que pedir un verbo y fijarse a qué conjugación corresponde. Luego preguntarnos en qué modo y tiempo queremos conjugarlo. En la figura 2 presentamos este diagrama de bloques.

Como primera instrucción de esta parte, también va a ser un CLS, porque nos aseguraremos así, que la pantalla no tenga "basura" que obstaculice la lectura de los mensajes de esta porción del programa.

Los números de línea no tienen porque seguir en forma creciente a la porción del programa anterior, pero sí es importante que los valores de las líneas no coincidan. Por ejemplo, comencemos desde la línea 1000. Copiemos a partir de la línea con este número del listado de la figura 3.

En la segunda línea pedimos la entrada de un verbo en infinitivo.

La línea 1030 es interesante pues separamos del verbo entrado, la raíz de la terminación.

Sabemos que la terminación, son sólo las dos últimas letras. Entonces, con la instrucción necesitaríamos una instrucción que tome las dos últimas letras. La instrucción MID$(X$,A,B) toma de la cadena de caracteres X$, los caracteres entre A y B.

Por eso, en la línea 1030, RAIZ$ representa a la raíz del verbo entrado y TERMI$ a la terminación de dicho verbo.

¿Qué significará la instrucción IF A OR B THEN...? Con sólo traducirla del inglés, significa que si pasa A o si pasa B, entonces... Esta instrucción se utiliza en la líne 1035 para distinguir a qué grupo pertenece el verbo entrado; si a los de primera conjugación, segunda o tercera.

Los datos que van a ser leídos por el programa, pero que no hagan falta entrar desde el teclado, se guardarán en sentencias DATAs. En los programas puede haber varias de estas sentencias de almacenamiento de datos.

Con la instrucción READ A$ de la línea 1038 lee un dato de estas sentencias DATA. Pero la lectura es secuencial, es decir que para leer el décimo dato habrá que leer o saltar los nueve anteriores.

En la línea 1037 encontramos la instrucción RESTORE 1039, que indica a la computadora posicionarse en la línea 1039 para leer datos. ¿Pero, por qué hace falta agragar esta instrucción?

Pues si el último dato leído fue, por ejemplo el quinto de la línea 1235 entonces al encontrar un READ leerá a partir del secto de la línea 1235.

En la línea 1038 encontramos la instrucción RESTORE 1039, que indica a la computadora posicionarse en la línea 1039 para leer datos.

¿Por qué hace falta agregar esta instrucción?

Porque si el último dato leído fue por ejemplo el quinto de la línea 1235 entonces al encontrar un READ leerá a partir del sexto dato de la línea 1235.

En la línea 1038 la instrucción FOR F=1 TO 6... NEXT ordena que se ejecuten 6 veces las instrucciones que se hallan entre el FOR y el NEXT.

La definición de esta instrucción es: FOR N=A TO B STEP C... NEXT. Esto quiere decir que se realizarán las instrucciones entre el FOR y el NEXT todas las veces que sean necesarias para que N (con valor inicial A), incrementándose C unidades, alcance o supere el valor de B. Esto, en computación se denomina "loop". La variable PER$(n) en realidad es un arreglo que almacena los valores leídos de la línea 1039 por el READ.

La línea 1050 nos pide una respuesta desde el teclado que puede ser un número entre 1 y 4 inclusive.

En la 1060 decidimos que si nuestra respuesta fue 1, pasaremos a la línea 1070, si fue 2 a la 1310, si en cambio fue 3 a la 1450, pero si contestamos 4, pasamos a la línea 1540.

Para realizar ésto, usamos el recurso ON N GOTO A,B,C... porque según sea el valor de N, salta a la línea A,B ó C.

Ya nos quedan pocas cosas por aprender. Por último vamos a ver algunos separadores de campo de la instrucción PRINT.

Vimos que las comas en un PRINT permiten comenzar en la primera columna o en la décimo quinta; éste es un separador de campo.

Otro es el punto y coma (;): indica que la siguiente impresión se efectuará pegada a la última realizada. Otro separador que realiza la misma operación es el signo "+", pero ésta se utiliza entre variables caracteres. Por ejemplo, PRINT A$+B$ imprimirá A$ y pegado imprimirá B$. Pero PRINT A+ imprimirá el resultado de sumar A con B.

El resto del programa no contiene instrucciones nuevas.

Pero sólo tenemos dos partes del programa y están "desconectadas". Con las líneas 700-990 unimos ambas rutinas sin usar instrucciones desconocidas.

Al programa se le pueden agregar infinitas variantes, pero las dejamos liberadas a la imaginación del lector. Como sugerencia, decimos que se le podría agregar la conjugación de algunos verbos irregulares como SER y ESTAR. Recomendamos mantener los números de líneas porque en números sucesivos ampliaremos el programa.

Una vez que tengamos copiado todo el listado de la figura 3, en ves de hacerlo correr con RUN, demos la orden con GOTO 700.

Esta instrucción indica que el comienzo del programa es la línea 700.

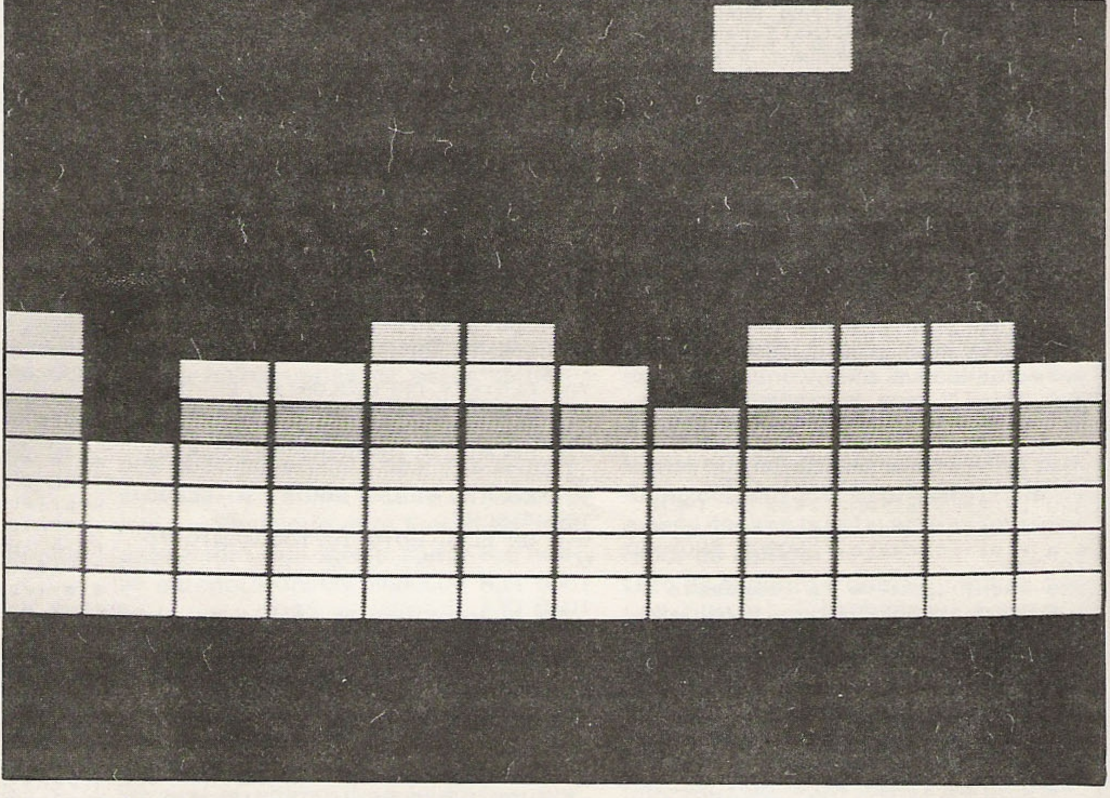
```
10 REM EXPLICACION DE LOS VERBOS
20 CLS:KEY OFF:COLOR 12,1
30 PRINT" Los accidentes verbale
s son:           MODO, TIEMPO, PERSO
NA y NUMERO."
40 PRINT:PRINT"MODO",,,," Indica
 las distintas actitudes del    qu
e habla:"
50 PRINT:PRINT:PRINT"- INDICATIV
O: expresa como real el       signi
ficado del verbo."
60 PRINT:PRINT TAB(5)"El cartero
 llamó a mi puerta."
70 PRINT:PRINT "- SUBJUNTIVO: ex
presa un deseo o una   duda."
80 PRINT:PRINT TAB(5)"Que el car
tero llame a mi puerta": GOSUB 59
0
90 PRINT:PRINT "- POTENCIAL: exp
resa posibilidad."
100 PRINT:PRINT TAB(5)"El carter
o llamaría a mi puerta."
110 PRINT:PRINT "- IMPERATIVO: e
xpresa pedido, orden,   invitació
n o consejo"
120 PRINT:PRINT TAB(5)"Que llame
, el cartero, a mi puerta.": GOSU
B 590
130 PRINT"Veremos si entendió lo
 que vimos      recién"
140 PRINT: PRINT"A qué modo corr
esponden las siguien- tes oracion
es:"
150 PRINT: PRINT:PRINT TAB(5)"*
Claudio escribiría una poesía":GO
SUB 610
160 IF X=3 THEN GOTO 180
170 BEEP:PRINT"Creo que no enten
dió bien la explica-ción, volvamo
s a leerla. Su respuestafue erron
ea, era en el modo POTENCIAL":GOS
UB 590: GOTO 30
180 PRINT: PRINT:PRINT TAB(5)"*
Claudio escribió una poesía":GOSU
B 610
190 IF X=1 THEN GOTO 210
200 BEEP:PRINT"Será mejor que vo
lvamos a leer la ex-plicación por
que la respuesta         correcta
era INDICATIVO.":GOSUB 590: GOTO
30
```

```
210 PRINT: PRINT:PRINT TAB(5)"*
Escribe, Claudio, una poesía":GOS
UB 610
220 IF X=4 THEN GOTO 240
230 BEEP:PRINT"Su respuesta no e
s satisfactoria, erael modo IMPER
ATIVO. Volvamos a leer  la explic
ación.":PRINT : GOSUB 590: GOTO 3
0
240 PRINT: PRINT:PRINT TAB(5)"*
Que Claudio escriba una poesía":G
OSUB 610
250 IF X=2 THEN PRINT:PRINT:GOSU
B 590:GOTO 270
260 BEEP:PRINT"Lástima que se ha
ya equivocado a estaaltura, pero
la respuesta correcta    era SUBJU
NTIVO.Tendrá que volver a    leer
la explicación.": PRINT: GOSUB 59
0: GOTO 30
270 PRINT"TIEMPO",,,," Indica cu
ándo se cumple la acción     verba
l:"
280 PRINT :PRINT"- Tiempos SIMPL
ES: presente perfecto,  pretérito
 imperfecto, pretérito        indef
inido, futuro imperfecto"
290 PRINT: PRINT"- Tiempos COMPU
ESTOS: pretérito          perfecto,
 pretérito anterior,          preté
rito pluscuanperfecto, futuro    p
erfecto":GOSUB 590
300 PRINT"Llegó el momento de sa
ber si aprendiólo que le acabo de
 explicar.": PRINT
310 INPUT"* Es el PRETERITO INDE
FINIDO simple.   (S/N)";X$
320 IF X$<>"s" AND X$<>"S" THEN
PRINT"Preste mas atención, volver
é a expli-cárselo.":GOSUB 590: GO
TO 270
330 PRINT:INPUT"* Es compuesto e
l FUTURO IMPERFECTO.   (S/N)";X$
340 IF X$<>"N" AND X$<>"n" THEN
PRINT"Preste mas atención, volver
é a expli-cárselo.":GOSUB 590: GO
TO 270
350 PRINT:INPUT"* Es compuesto e
l PRETERITO ANTERIOR.  (S/N)";X$
360 IF X$<>"s" AND X$<>"S" THEN
PRINT"Preste mas atención, volver
é a expli-cárselo.":GOSUB 590: GO
TO 270
370 INPUT"* Es el PRETERITO PLUS
CUANPERFECTO       compuesto.    (S/
N)";X$
380 IF X$<>"s" AND X$<>"S" THEN
PRINT"Preste mas atención, volver
é a expli-cárselo.":GOSUB 590: GO
TO 270
390 CLS:PRINT"PERSONA",,,," Indi
ca quién realiza la acción
verbaly puede ser:"
400 PRINT: PRINT"1- yo","1- noso
tros","2- tú","2- vosotros","3- é
l","3- ellos"
410 PRINT: PRINT:PRINT"NUMERO",,
,," Indica si el verbo se refiere
 a uno   o a muchos, SINGULAR o PL
URAL res-    pectivamente.":GOSUB
590
420 PRINT "Haré algunas pregunta
s para saber si entendió este últ
imo punto."
430 INPUT " *Cuál es la primera
persona del        singular";X$
440 IF X$<>"YO" AND X$<>"yo" THE
N PRINT"Será mejor que le vuelva
a explicar  esto último porque pa
rece que no me  entendió.":GOSUB
590: GOTO 390
450 INPUT " *Cuál es la tercera
 persona del       singular";X$
460 IF X$<>"él" AND X$<>"EL" THE
N PRINT"Será mejor que le vuelva
a explicar  esto último porque pa
rece que no me  entendió.":GOSUB
590: GOTO 390
470 INPUT " *Cuál es la segunda
 persona del       plural";X$
480 IF X$<>"vosotros" AND X$<>"V
OSOTROS" THEN PRINT"Será mejor qu
e le vuelva a explicar  esto últi
mo porque parece que no me   enten
dió.":GOSUB 590: GOTO 390
500 INPUT " *Cuál es la tercera
 persona del       plural";X$
510 IF X$<>"ellos" AND X$<>"ELLO
S" THEN PRINT"Sera mejor que le v
uelva a explicar  esto último por
que parece que no me  entendió.":
GOSUB 590: GOTO 390
520 CLS:PRINT"Al conjugar un ver
bo se lo debe deciren forma orden
ada en los distintos    modos, tie
mpos, personas y números."
530 PRINT: PRINT"Los verbos se c
lasifican según sus   conjugacion
es y esto depende de las   termina
ciones verbales."
550 PRINT:PRINT"CONJUGACIONES","
 TERMINACIONES":PRINT
560 PRINT"primera","  AR","segun
da","  ER","tercera","  IR"
570 PRINT:PRINT"Como por ejemplo
 los verbos amAR,     tenER y part
IR que corresponden a la primera,
 segunda y tercera conju-    gaci
ón respectivamente.":GOSUB 590
580 LOCATE 0,13: PRINT "Aquí ter
minó la explicación de los   verb
os.": GOSUB 590: GOTO 700
590 LOCATE 0,22: PRINT "Presione
 alguna tecla para continuar"
600 IF INKEY$="" THEN 600 ELSE C
LS:RETURN
610 PRINT: PRINT "1- Indicativo"
,"2- Subjuntivo","3- Potencial","
4- Imperativo"
620 INPUT "Qué numero elige: ";X
: RETURN
700 CLS:KEY OFF:COLOR 10,1
710 LOCATE 0,8:PRINT "1- Explica
cion."
720 LOCATE 0,10:PRINT "2- Conjug
acion."
980 LOCATE 0,22: INPUT "SU ELECC
ION ";ELEC
990 ON ELEC GOTO 10,1000:GOTO 70
0
1000 REM COJUGAR VERBOS
1010 CLS
1020 INPUT"Entre el verbo que qu
iere conjugar ";X$
1025 IF X$="FIN" OR X$="fin" THE
N GOTO 700
1030 RAIZ$=MID$(X$,1,LEN(X$)-2):
 TERMI$=MID$(X$,LEN(X$)-2+1,LEN(X
$))
1035 IF TERMI$="AR" OR TERMI$="a
r" THEN T=1 ELSE IF TERMI$="ER" O
R TERMI$="er" THEN T=2 ELSE T=3
1036 IF T=1 THEN T$="ado" ELSE T
$="ido"
1037 RESTORE 1039
1038 FOR F=1 TO 6: READ A$: PER$
(F)=A$:NEXT
1039 DATA yo,tú,él,nosotros,voso
tros,ellos
1040 LOCATE 0,5: PRINT"1- Indica
tivo","2- Subjuntivo","3- Potenci
al","4- Imperativo"
1050 LOCATE 0,7:PRINT: INPUT"En
qué modo quiere conjugar al verbo
:";M
1060 ON M GOTO 1070,1310,1450,15
40
1070 REM INDICATIVO
1080 CLS: PRINT "1- Presente",,"
2- Pretérito imper.","3- Pretérit
o indef.","4- Futuro imper.","5-
Pretérito perf.","6- Pretérito pl
uscuan.","7- Pretérito anterior",
"8- Futuro perf."
1090 LOCATE 0,13: INPUT "En qué
tiempo verbal lo quiere:";TIE: CL
S
1210 IF TIE>4 THEN GOTO 1250
1215 RESTORE 1235: IF T=1 AND TI
E=1 THEN 1230
1220 FOR F=1 TO 18*(TIE-1)+6*(T-
1): READ A$: NEXT
1230 FOR F=1 TO 6: READ A$: PRIN
T PER$(F), RAIZ$+A$: NEXT: GOSUB
1970
1235 DATA o,as,a,amos,áis,an,o,e
s,e,emos,éis,en,o,es,e,imos,ís,en
,aba,abas,aba,ábamos,abais,aban,í
a,ías,ia,íamos,íais,ían
1236 DATA ía,ías,ía,íamos,íais,í
an,é,aste,ó,amos,asteis,aron,í,is
te,ió,imos,isteis,ieron,í,iste,ió
,imos,isteis,ieron
1237 DATA aré,arás,ará,aremos,ar
éis,arán,eré,erás,erá,eremos,eréi
s,erán,iré,irás,irá,iremos,iréis,
irán
1250 IF TIE=5 THEN RESTORE 1270
ELSE IF TIE=6 THEN RESTORE 1280 E
LSE IF TIE=7 THEN RESTORE 1290 EL
SE RESTORE 1300
1260 FOR F=1 TO 6: READ A$: PRIN
T PER$(F)TAB(11)A$;" ";RAIZ$+T$:N
EXT:GOSUB 1970
1270 DATA he,has,ha,hemos,habéis
,han
1280 DATA había,habías,había,hab
íamos,habíais,habían
1290 DATA hube,hubiste,hubo,hubi
mos,hubisteis,hubieron
1300 DATA habré,habrás,habrá,hab
remos,habréis,habrán
1310 REM SUBJUNTIVO
1320 CLS: PRINT "1- Presente",,"
2- Pretérito imper.","3- Futuro i
mper.","4- Pretérito perf.","5- P
retérito pluscuan.","6- Futuro pe
rf."
1330 LOCATE 0,13: INPUT "En qué
tiempo verbal lo quiere:";TIE: CL
S
1350 IF TIE>3 THEN 1400
1355 RESTORE 1380: IF T=1 AND TI
E=1 THEN 1370
1360 FOR F=1 TO 18*(TIE-1)+6*(T-
1): READ A$: NEXT
1370 FOR F=1 TO 6: READ A$: PRIN
T PER$(F), RAIZ$+A$: NEXT: GOSUB
1970:GOTO 1000
1380 DATA e,es,e,emos,éis,en,a,a
s,a,amos,áis,an,a,as,a,amos,áis,a
n,ara o ase,aras o ases,ara o ase
,áramos o ásemos,arais o aseis,ar
an o asen
1382 DATA iera o iese,ieras o ie
ses,iera o iese,iéramos o iésemos
,ierais o ieseis,ieran o iesen,ie
ra o iese,ieras o ieses,íera o ie
se,iéramos o iésemos,ierais o ies
eis,ieran o iesen
1383 DATA are,ares,are,áremos,ar
eis,aren,iere,ieres,iere,iéremos,
iereis,ieren,iere,ieres,iere,iére
mos,iereis,ieren
1400 IF TIE=4 THEN RESTORE 1420
ELSE IF TIE=5 THEN RESTORE 1430 E
LSE RESTORE 1440
1410 FOR F=1 TO 6: READ A$: PRIN
T PER$(F)TAB(10)A$;" ";RAIZ$+T$:N
EXT:GOSUB 1970
1420 DATA haya,hayas,haya,hayamo
s,hayáis,hayan
1430 DATA hubiera o hubiese,hubi
eras o hubieses,hubiera o hubiese
,hubiéramos o hubiésemos,hubierai
s o hubieseis,hubieran o hubiesen
1440 DATA hubiere,hubieres,hubie
re,hubiéremos,hubiereis,hubieren
1450 REM POTENCIAL
1460 CLS: PRINT"1- Simple","2- C
ompuesto"
1470 LOCATE 0,13:PRINT:INPUT"En
qué tiempo quiere conjugarlo ";TI
E: CLS
1480 IF TIE=2 THEN 1515
1485 RESTORE 1510: IF T=1 AND TI
E=1 THEN 1500
1490 FOR F=1 TO 6*(T-1):READ A$:
 NEXT
1500 FOR F=1 TO 6: READ A$: PRIN
T PER$(F), RAIZ$+A$: NEXT: GOSUB
1970
1510 DATA aría,arías,aría,aríamo
s,aríais,arían,ería,erías,ería,er
íamos,eríais,erían,iría,irías,irí
a,iríamos,iríais,irian
1515 RESTORE 1530
1520 FOR F=1 TO 6: READ A$: PRIN
T PER$(F)TAB(11)A$;" ";RAIZ$+T$:N
EXT:GOSUB 1970:GOTO 1000
1530 DATA habría,habrías,habría,
habríamos,habríais,habrían
1540 REM IMPERATIVO
1550 CLS :RESTORE 1600:IF T=1 TH
EN 1570
1560 FOR F=1 TO 5*(T-1):READ A$:
 NEXT
1570 FOR F=1 TO 5: READ A$
1580 PRINT RAIZ$+A$ TAB(10)PER$(
F+1)
1590 NEXT:GOTO 1970:GOTO 1000
1600 DATA a,e,emos,ad,en,e,a,amo
s,ed,an,e,a,amos,id,an
1970 LOCATE 0,22:PRINT "Presione
 alguna tecla para continuar"
1980 IF INKEY$="" THEN 1980 ELSE
 GOTO 1000
```



LADRI

Clase: juego



Les presentamos aquí una interesante versión de un juego con el que habrán pasado horas y gastado más de una ficha en Mar del Plata.

Fue implementada por uno de nuestros colaboradores y amigo Gustavo Faigenbaum, docente del Club de usuarios de MSX, instalado en el CEDEI.

Dejemos pues que él mismo nos explique su programa:

LADRI es una versión simplificada del ya clásico "Break Out".

Como se ve, el listado es corto e íntegramente en BASIC.

Si bien esto puede quitarnos un poco de definición y velocidad, no debemos tomar al listado de LADRI sólo como un juego, sino como una demostración de lo mucho que se puede hacer en poco tiempo y con algo de imaginación, programando en BASIC-MSX.

Notemos que para distinguir el choque de la pelota contra la raqueta se usa la instrucción "ON SPRITE GOSUB". En cambio para localizar el choque contra los límites de la pantalla se usa la función "POINT".

Estructura del programa

10-330: Inicialización de variables, sprites e imagen de pantalla.

340: Movimiento de la paleta.

350-400: Movimiento de la pelotita.

410-490: Destrucción de ladrillos. En la línea 470 se contempla la posibilidad de terminar con una pared.

500-530: Subrutina de choque de la paleta con la pelota.

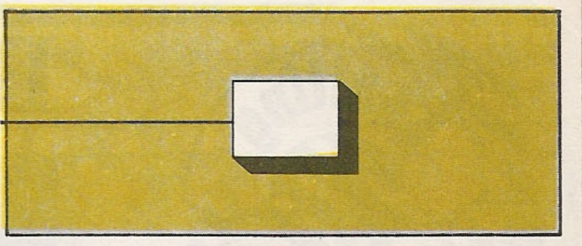
540-570: Subrutina de pérdida de la pelotita.

580-610: Subrutinas auxiliares de cálculo de dirección de la pelota.

620-790: Finalización del juego y cálculo del record.

```
10 REM******* LADRI ************
20 REM**POR GUSTAVO FAIGENBAUM**
30 COLOR 11,1,1
40 N=0
50 J=5
60 SCREEN2,3
70 RESTORE
80 P$="":B=0
90 SPRITE ON
100 ON SPRITE  GOSUB 490
110 FOR A=0 TO 27
120 P$=P$+CHR$(255)
130 NEXT
140 SPRITE$(0)=P$
150 P$=""
160 FOR P=0 TO 7
170 READ A
180 P$=P$+CHR$(A)
190 NEXT
200 CLS
210 SPRITE$(2)=P$
220 DATA 0,  0,60 ,60 ,60,60 ,0,
0
230 W=N+3
240 FOR A=20TO96STEP 12
250 FOR P=0 TO 240 STEP 21
260 LINE (P,A)-STEP (19,10),11-A
/11,BF
270 NEXT
280 NEXT
290 PUTSPRITE0,(60,173),12
300 IF STRIG(0)<>-1THEN 300
310 BEEP
320 X=60:Z=3:Y=110:V=1.5:GOSUB 5
70:GOSUB 590
330 IF STICK(0)=3 AND X<212THENX
=X+8:PUTSPRITE0,(X,173),12ELSEIFS
TICK(0)=7 AND X>0THENX=X-8:PUT SP
RITE 0,(X,173),12
340 IF Y<-WTHEN W=-W:V=V* INT(1.
08+RND(1)):BEEP:GOSUB 590
350 IF Z<-VTHEN V=ABS(V):BEEP:GO
SUB 570
360 IF Z+V>232THEN V=ABS(V)*-1:B
EEP:GOSUB 570
370 Y=Y+W:Z=Z+V
380 PUT SPRITE2,(Z,Y),14
390 IFY>188THEN530
400 IFABS(POINT(Z+C,Y+D))=1THEN3
30
410 A=POINT(Z+C,Y+D)
420 PAINT(Z+C,Y+D),1,1
430 BEEP
440 W=-W:GOSUB 590
450 B=B+A
460 IF B=456 THEN N=N+1:SCREEN 1
:PRINT"Lo felicito .Puntaje hasta
    el momento:"  N   * 456  ,"Pr
eparese para nueva serie" :FOR I=
1 TO 3000:NEXT:SCREEN 2,3:GOTO 60
470 IF A+N>ABS(W)THEN W=SGN(W)*(
A+N-1):V= V*INT(1.05+RND(1))
480 GOTO 330
490 IF Y<170 THEN W=-ABS(W):ELSE
 V=-V
500 BEEP
510 Z=Z+V:Y=Y+W:PUT SPRITE2,(Z,Y
),14:GOSUB 590:GOSUB 570
520 RETURN
530 W=N+2
540 J=J-1
550 IF J=0THEN 610
560 GOTO 290
570 IF V>0 THEN C=8:ELSE C=0
580 RETURN
590 IF W>0 THEN D=8:ELSE D=0
600 RETURN
610 SCREEN 1
620 PRINT"PUNTAJE FINAL ";N*456+
B
630 IF N*456+B>ME THEN PRINT"LO
FELICITO, USTED HA BATIDO EL RECO
RD.":LINE INPUT "¿COMO SE LLAMA?
";ME$:ME=N*456+B
640 PRINT
650 PRINT"RECORD ";ME;" POR ";ME
$
660 PRINT"¿DESEA VOLVER A JUGAR?
        (s/n)"
670 A$=INPUT$(1)
680 IF A$="n"ORA$="N"THEN END
690 IF A$="S"ORA$="s"THEN 40
700 GOTO 670
```

CONCURSO DE PROGRAMAS



auspiciado por TELEMATICA S.A. que proveerá los siguientes Premios:

P R I M E R P R E M I O

Un periférico (a elección entre un monitor, una diskettera y una impresora). **Una Beca** para trabajar en el Departamento de Investigación y Desarrollo de Telemática S.A.

S E G U N D O P R E M I O

Un periférico (a elección entre un monitor, una diskettera y una impresora).

En caso de que el ganador no pueda utilizar la beca será ofrecida a quien obtenga el segundo premio, y si este tampoco pudiera aprovecharla se otorgará a alguno de los participantes del certamen que se hubiera destacado.

ESPECIAL Entre los programas recibidos, algunos de ellos podrán ser editados por Prosoft, reconociéndose los derechos de autor.

J U E G O S :

Temas: ● **TRUCO.** Premiaremos al programador que logre la mejor versión de este tradicional juego de salón.

● **DE INTELIGENCIA.** Los juegos que nos inspiren podrán ser "El Estanciero" o temas originales que sigan la línea

E D U C A T I V O S :

Tema: ● **LIBRE.**

U T I L I T A R I O :

Tema: ● **INTELIGENCIA ARTIFICIAL.** Se trata de lograr un programa que realmente nos haga discutir sobre el límite entre la simulación y la inteligencia de la computadora.

Obviamente la única forma de lograr esto será siguiendo las reglas propuestas de la Inteligencia artificial. Por esto se considerarán para premiar esta categoría, además de las reglas detalladas más abajo, la capacidad de auto-aprendizaje del programa, el nivel de inferencia del mismo, la capacidad y modo de almacenamiento de su base de datos, y principalmente su analizador sintáctico dado que hasta el momento no se ha logrado uno que dé suficiente credibilidad de que estamos frente a un ser racional que entiende nuestro idioma y no frente a una máquina a la que debemos hablarle con verbos en infinitivo al mejor estilo Tarzán (sin menospreciar a este último).

PROFESIONAL O GESTION:

Tema: ● **LIBRE.** Dentro de este tema podrán figurar programas de las más diversas aplicaciones.

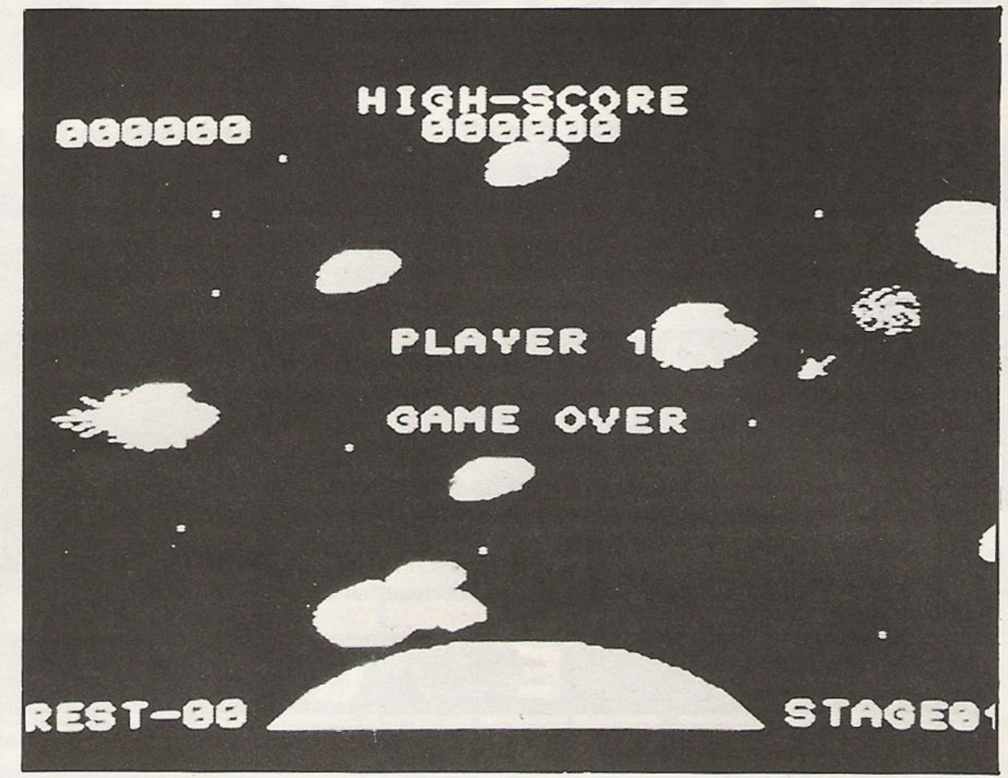
BASES: No sólo será indispensable que el programa enviado funcione correctamente sino que además debe cumplir con ciertas reglas.

- *Programación estructurada en bloques fácilmente diferenciables.*
- *Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de línea que los identifiquen).*
- *Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.*
- *Las variables y/o direcciones de memoria utilizados también se deben incluír en esta documentación.*
- *Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.*
- *Calidad y originalidad de gráficos, sonidos y pantallas de menú.*
- *El programa debe ajustarse a alguno de los temas propuestos más arriba. Esto es ELIMINATORIO.*

Los trabajos deberán enviarse antes del 30 de noviembre próximo (cierre del certámen) a: Paraná 720, piso 5º, (1017) Capital Federal.

Raiting Soft

BATTLE CROSS



CREATIVIDAD: **5**
PRESENTACION: **2**
ATRACCION: **7**
GRAFICOS: **8**
SONIDO: **6**
TIPO: **ENTRETENIMIENTO**
PRODUCE: **Bitgame**

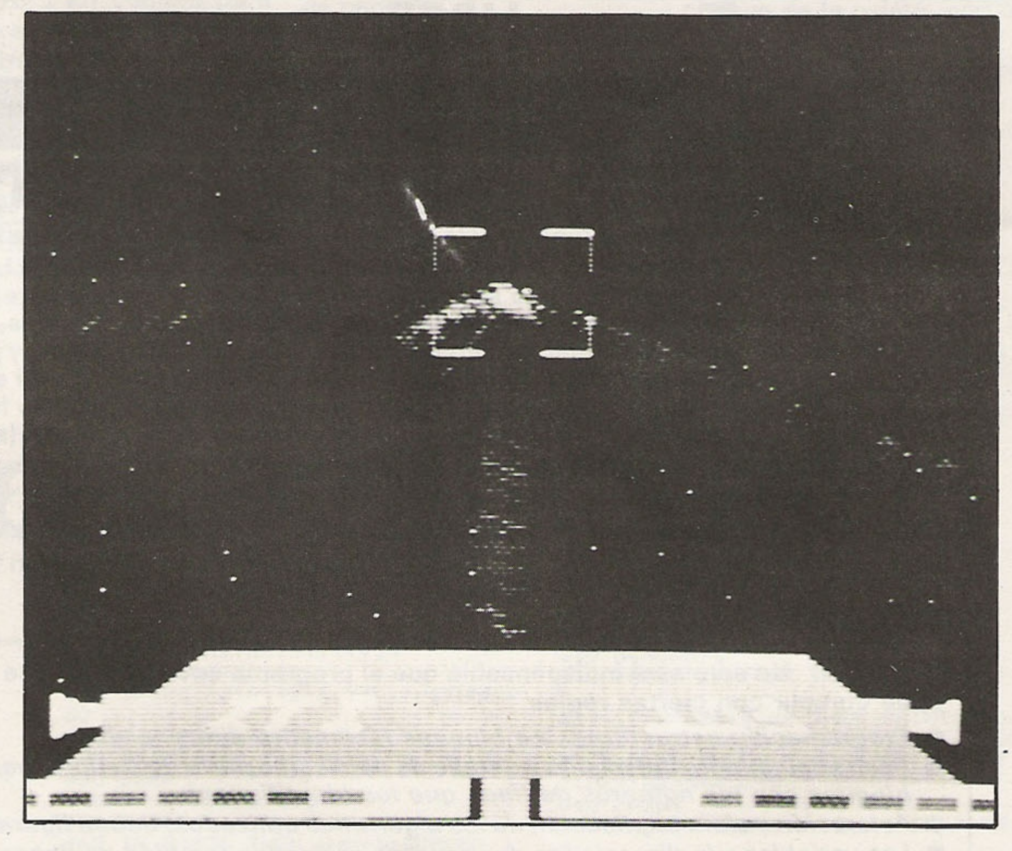
Orbitando en torno de algún planeta, nuestra nave debe defenderse de otras naves y seres extraños. Esto, además de esquivar lluvias tupidas de meteoros gigantescos que invaden nuestra órbita.

El juego es simple y tradicional. No se ha innovado nada, pero en conjunto resulta muy atractivo por sus gráficos y la acción que desarrolla y que nos obliga a mantener.

Para los amantes de los juegos de acción galáctica tradicional, tipo PARSEC, este programa no los defraudará. Bienvenidos al hostil universo.

TOEMI

CREATIVIDAD: **6**
PRESENTACION: **4**
ATRACCION: **9**
GRAFICOS: **8**
SONIDO: **7**
TIPO: **ENTRETENIMIENTO**
PRODUCE:´ **Bitgame**





Este juego puede enrolarse dentro de los de acción espacial, en los que el desarrollo transcurre desde la cabina de una nave intergaláctica.

Un poderoso laser es la única arma disponible para defendernos de los ataques de los objetos voladores no identificados.

Cuatro grandes reactores son los encargados de propulsar y guiar a nuestro artefacto por el espacio infinito.

Estos son comandados desde las hermosas teclas azules de cursor.

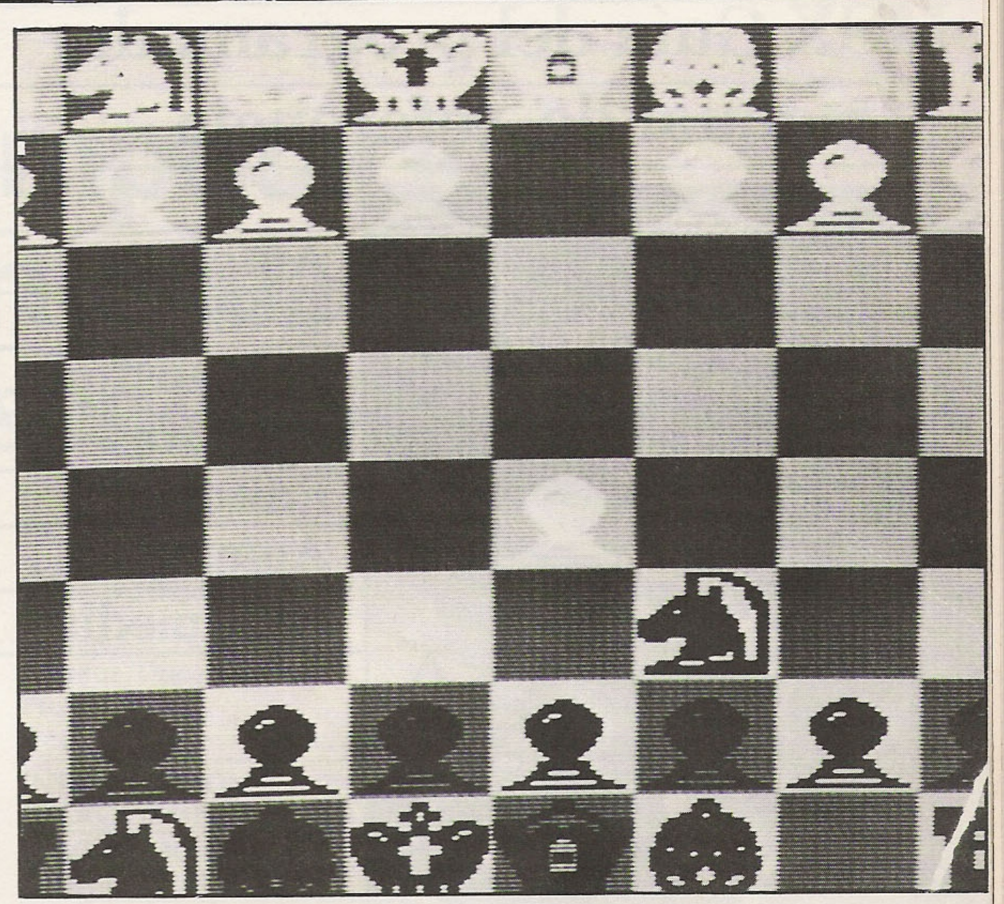
El desarrollo, si bien no es original del autor, no lo desmerece en absoluto pues la idea base ha sido complementada por buenas y originales rutinas.

Entre ellas, una que nos felicita luego de haber cumplido parte de la misión, muestra una excelente escena en la que un trasbordador espacial abre su bodega para proveernos del indispensable combustible, y en nuestro visor aparece acompañándolo lentamente un enorme planeta.

En resumen, es un buen juego, que nos transportará efectivamente hacia la despreocupación, por lo menos momentánea.

AJEDREZ

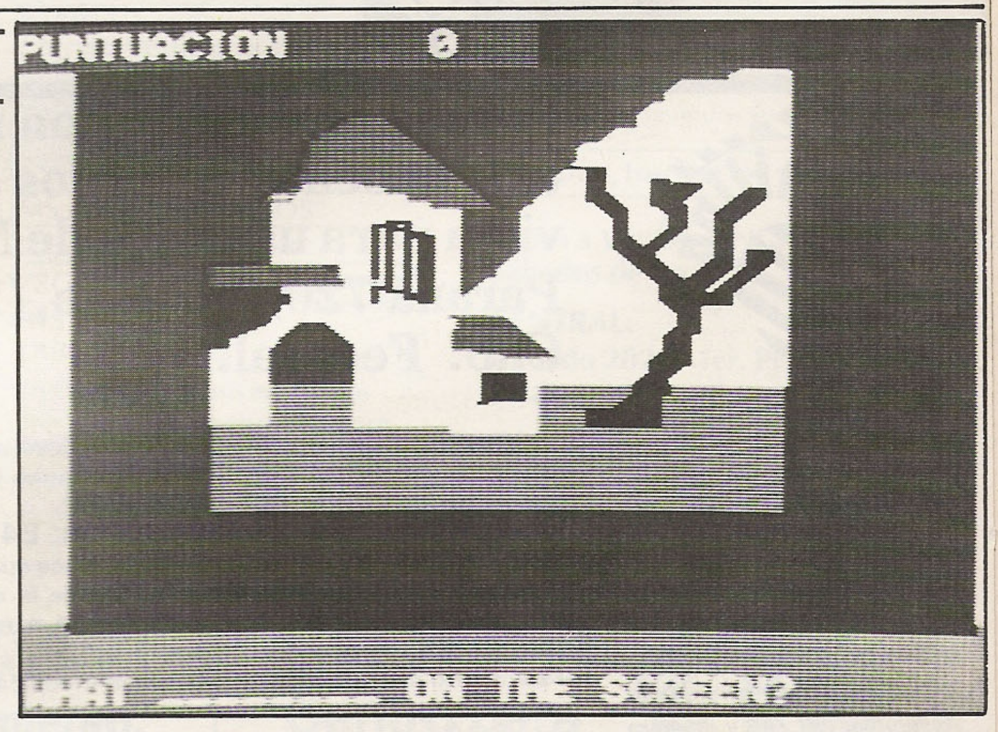
CREATIVIDAD: **–**
PRESENTACION: **5**
ATRACCION: **10**
GRAFICOS: **8**
SONIDO: **7**
TIPO: **ENTRETENIMIENTO**
PRODUCE: **Bitgame**



Este es un gran desarrollo, hecho sin duda con sendos conocimientos de las reglas de este juego o deporte (para no entrar en ese dilema), que realmente atrapará a los amantes de este eterno pasatiempo: El Ludo. —¿Qué...?, ah sí, El Ajedrez. Perdón.
En sí, el programa no defraudará a los exigentes y expertos jugadores ni en calidad gráfica, ni en la oposición que la máquina ofrece en sus más altos niveles.
Así, su adquisición es una buena idea, para desperdiciar el tiempo inteligentemente.

INGLES I

CREATIVIDAD: **8**
PRESENTACION: **7**
ATRACCION: **8**
GRAFICOS: **7**
SONIDO: **5**
TIPO **EDUCATIVO**
PRODUCE: **Bitgame**





Si alguna vez tuvimos la intención de buscar un juego que permitiera practicar las estructuras básicas del idioma inglés y al mismo tiempo entretenernos, este soft se aproxima bastante a su necesidad.
Los dos lados del cassette tienen preguntas para practicar. El lado A comienza con palabras básicas que a medida que avanzamos en el juego, se complican pues las respuestas deberán ser frases como HASN'T GOT por ejemplo.
Las preguntas que deberemos responder son sobre las distintas pantallas que se nos presenta.
Los gráficos tienen una combinación atractiva de colores que producen pantallas llamativas. Lamentablemente este programa no cuenta con efectos sonoros sobresalientes.
Seremos evaluados con treinta y ocho preguntas. Al finalizar el entretenimiento, obtendremos el resultado del interrogatorio.
El lado B del cassette es un poco más entretenido, aunque también más difícil. Nos encontramos en el interior de un castillo del que debemos tratar de escapar. Pero como primera dificultad, hay un extraterrestre que nos persigue y está identificado con el color verde. Este extraterrestre nos perseguirá constantemente. Y si consigue atraparnos tres veces, nos matará.
Pero lo más interesante de ese software es que nos hará adivinanzas que deberemos responder si queremos escapar.
Contamos con algunos recursos que nos ayudarán a huir del extraterrestre como, por ejemplo, transformarnos en invisibles.
Al comienzo del programa tenemos las explicaciones del manejo del juego.

MSX Guía del programador y manual de referencia



Autor:
T. Sato
P. Mapstone
I. Muriel
Editorial:
Anaya
Multimedia

Este pequeño libro de 702 páginas apenas, es como la gran biblia del MSX. Tiene casi todo lo que tienen los demás libros de la misma clase, es decir que es una especie de compendio de todos los demás.

Por otra parte, no escapa a la temática que se usa en ellos: enseña clara y eficientemente cada uno de los puntos destacables y originales que el sistema MSX posee.

Según la misma contratapa, el libro cubre todo lo que se necesita saber sobre MSX, desde el BASIC al Sistema Operativo, de forma exhaustiva pero con un estilo conciso, estando eficazmente estructurado para que se pueda utilizar de forma práctica como manual de consulta y referencia constantes o para aprender o programar y manejar con destreza cualquier micro MSX.

A pesar de esto, éste no es un manual para aprender a programar en BASIC. Posibilita además conocer el funcionamiento y manejo de los chips que habitan dentro de la carcasa de la máquina.

Al final del libro se describe detalladamente el sistema operativo MSX: instrucciones RST, puntos de entrada y salida, BIOS y vectores.

En fin, es un libro muy completo y recomendable como obra de consulta permanente.



Para comunicarse con nosotros deben escribirnos a Revista para usuarios de MSX, Paraná 720, 5to. Piso, (1017) Cap. Federal.

Load MSX direcciones

Ante todo quiero felicitarlos por su publicación, me parece fantástica. Actualmente estoy comenzando a aprender el Assembler del Z80 y quisiera acceder desde este lenguaje al manejo de información de cassette.

Alicia Colombo. La Plata.

Load MSX

Agradecemos las felicitaciones y esperamos seguir satisfaciendo las necesidades de nuestros lectores.

Con respecto al pedido, en próximas entregas veremos este tema detalladamente, pues es amplio y requiere de ciertas consideraciones previas.

De todos modos aquí van algunas de las posiciones de memoria que corresponden al comienzo de las rutinas que se encargan de controlar este periférico.

Así que a cada una de ellas deberemos llamarlas con un CALL.

Dirección: E1

Función: Hace que el motor comience a funcionar y lee el encabezamiento del programa.

Hay que tener en cuenta que una llamada a esta rutina, modificará todos los registros.

Esta rutina devolverá un 1 en el bit de acarreo si se pulsa CTRL-STOP.

Dirección: E4

Función: Hace que los datos entren desde la cinta. Este dato quedará en el acumulador.

También modifica los registros.

Dirección: E7

Función: Detiene la lectura de datos desde el cassette.

Dirección: EA

Función: Graba el encabezamiento de un programa. Si el encabezamiento es corto, el acumulador debe contener un 0 antes de llamarlo. Otro valor significará que el encabezamiento es largo.

Dirección: ED

Función: Graba el dato contenido en el acumulador en cinta.

Dirección: F0

Detiene el flujo de datos desde la máquina hacia el grabador.

Dirección: F3

Función: Acciona el motor del grabador.

Peek

He leído en algunos programas que han publicado, el uso de la instrucción PEEK y como no tengo aún conocimientos sólidos de programación, que espero algún día tener, me gustaría que me expliquen para qué se utiliza en los programas.

Jorge Urdanivia. Capital

Load MSX

Esta instrucción se utiliza cuando queremos leer alguna posición de memoria de nuestra computadora. La forma de esta instrucción (como lo indica el manual) es: PEEK (n) y leer el valor existente en la posición n.

Computación, una oportunidad para que todos enseñen y aprendan.

Un lugar para

desarrollar el pensamiento.

descubrir una vocación.

manejar lenguajes de computación.

comprender los múltiples usos de un computador.

capacitar y perfeccionar al docente.

incorporar los avances tecnológicos.

que el profesional domine el uso de nuevas herramientas.

que los padres se reencuentren con sus hijos.

"No se trata solamente de adquirir en forma puntual conocimientos definitivos, sino prepararse a elaborar a lo largo de toda la vida, un saber en constante evolución y de aprender a ser."

UNESCO

Actividades '86

Para Niños, Adolescentes, Adultos, Docentes, Profesionales y Establecimientos educativos.

- INTRODUCCION A MICROCOMPUTADORES
- DIAGRAMACION ESTRUCTURADA
- LOGO
- BASIC
- COLOR - SPRITE - SONIDO
- COBOL
- PASCAL
- ASSEMBLER
- MS - DOS Y MSX - DOS
- D BASE II - MULTIPLAN
- PROCESADOR DE LA PALABRA
- INSTALACION DE LABORATORIOS

en Establecimientos educativos con formación de multiplicadores y apoyo a la comunidad.

Cómo?

- Taller en grupos de 12 a 15 personas.
- Clases de 2 horas diarias.
- 2 ó 3 alumnos por equipo.
- Equipos disponibles para prácticas adicionales en horarios libres.
- Becas rentadas en el Departamento de investigación y desarrollo de Talent MSX.
- Becas rentadas para docentes en Laboratorios de Establecimientos Educativos.

Informes, Inscripción y Cursos

Lunes a Viernes de 8 a 22 hs.
Sábados de 8 a 13 hs.

CENTRAL:
Cabildo 2027 - 1er. Piso y Juramento

FILIALES:
Centro: Esmeralda 320 - 5º P.
Lanús: Caaguazú 2186 - L. Este

Talent MSX
Inteligencia en crecimiento.

Cedei
Centro para el desarrollo de la inteligencia.

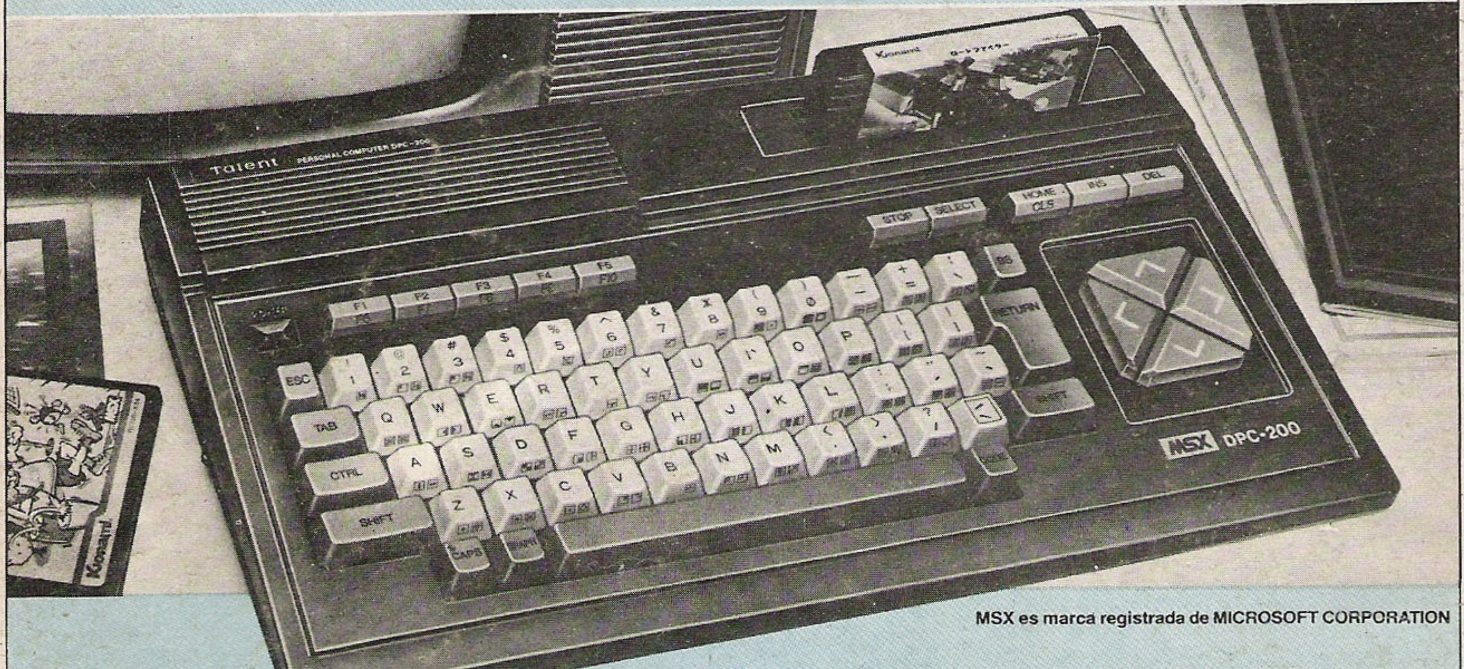
Descubramos y construyamos juntos los caminos que nos permitirán el uso inteligente de los productos de la creatividad humana.

diálogo - *D.P.*

Títulos no oficiales

A la computadora personal
Talent MSX
nada le es imposible

dialogo - D.P.



MSX es marca registrada de MICROSOFT CORPORATION

Porque gracias a la norma internacional MSX, la TALENT MSX trasciende sus propios límites.
Hasta ahora, cuando usted compraba una computadora personal de cualquier marca, quedaba automáticamente desconectado del resto del mundo de la computación. Porque los distintos equipos y sistemas no eran compatibles entre sí.
Hasta que dos grandes empresas de informática, la Microsoft Corp. de EE.UU. y la ASCII del Japón se pusieron de acuerdo para crear una norma standard: la MSX. Que se expandió también rápidamente en Europa. Y que hoy TALENT presenta por primera vez en la Argentina.
Mientras que la mayoría de las computadoras de su tipo que se ofrecen en el mercado nacional, han sido discontinuadas por obsoletas en sus lugares de origen, TALENT MSX tiene casi ilimitadas posibilidades de desarrollo. Porque la norma MSX es en todo el mundo inteligencia en crecimiento.
La TALENT MSX pone a su disposición un mundo de software para elegir. Y con la incorporación de todos sus periféricos llega a ser una auténtica computadora profesional.

UTIL

Su poderoso sistema operativo MSX permite el acceso a todo tipo de procesamiento de datos:

- Planillas de calculo.
- Procesadores de palabra.
- Graficos de negocios.
- Bases de datos (d Base II, etc.)
- Contabilidad general, sueldos, y jornales, costos, etc., desarrollados bajo CP/M en Basic, Cobol, Pascal o C.

Con la posibilidad de conexion a linea telefonica permite la transferencia y consulta de datos entre computadoras personales, profesionales o bancos de datos.
La grabacion de archivos es en formato MS-DOS, haciendola compatible con las computadoras profesionales.

DIDACTICA

Dispone de tres lenguajes para la enseñanza de computacion: LOGO como lenguaje de induccion para los mas chicos. Lenguaje de Programacion en castellano, para todos los que quieran aprender a programar sin conocimientos previos. Y Basic MSX como lenguaje profesional. Mas una amplia variedad de perifericos como el Mouse, Lapiz Optico, Tableta grafica, Track-ball, etc.

DIVERTIDA

La mas genial para Video-Juegos. Por la amplisima biblioteca de programas **–todos nuevos–** de la norma MSX en el mundo. Y ademas, el Basic MSX permite al usuario generar sus propios juegos con un manejo tan simple, como solo TALENT MSX puede ofrecer.

CARACTERISTICAS TECNICAS

- Memoria principal 64 KB ampliable hasta 576 KB.
- Memoria de video: 16 KB RAM.
- ROM incorporada de 32 KB con el MSX-Basic de Microsoft.
- Graficos completos, hasta 32 sprites y 16 colores simultaneos.
- Generador de sonido de 3 voces y 8 octavas.
- Conexion para cualquier grabador.
- Interfaz para salida impresora paralela.
- Conectores para cartuchos y expansiones.
- Fuente para 220 V y modulador PAL-N incorporado.

DISTRIBUIDORES AUTORIZADOS: CAPITAL FEDERAL: AMATRIX, Bolivar 173 - ARGECINT, Av. de Mayo 1402 - BAIDAT COMPUTACION, Juramento 2349 - COMPUPRANDO, Av. de Mayo 965 - COMPUSHOP, Córdoba 1464 - COMPUTIQUE, Córdoba 1111, E. P. - COMPUTRONIC, Viamonte 2096 - CP67 CLUB, Florida 683, L. 18 - DALTON COMPUTACION, Cabildo 2283 - ELAB, Cabildo 730 - MICROSTAR, Callao 462 - Q.S.P., Bartolomé Mitre 864 - SERVICIOS EN INFORMATICA, Paraná 164 - DISTRIBUIDORA CONCALES, Tucumán 1458 - MICROMATICA, Av. Pueyrredón 1135 - **ACASSUSO:** MICROSTAR ACASSUSO, Eduardo Costa 892 - **AVELLANEDA:** ARGOS, Av. Mitre 1755 - **BOULOGNE:** COMPUTIQUE CARREFOUR, Bernardo de Irigoyen 2647 - **CASTELAR:** HOT BIT COMPUTACION, Carlos Casares 997 - **LANUS:** COMPUTACION LANUS, Caaguazú 2186 - **LOMAS DE ZAMORA:** ARGESIS COMPUTACION, Av. Meeks 269 - **MARTINEZ:** VIDEO BYTE, Hipólito Yrigoyen 32 - **RAMOS MEJIA:** MANIAC COMPUTACION, Rivadavia 13734 - **SAN ISIDRO:** FERNANDO CORATELLA, Cosme Beccar 249 - **VICENTE LOPEZ:** SERVICIOS EN INFORMATICA, Av. del Libertador 882 - **BAHIA BLANCA:** SERCOM, Donado 327 - SUMASUR, Alsina 236 - **LA PLATA:** CADEMA, Calle 7 Nº 1240 - CERO-UNO INFORMATICA, Calle 48 Nº 529 - **MAR DEL PLATA:** FAST, Catamarca 1755 - **NECOCHEA:** CAFAL, Calle 57 Nº 2920 - SERCOM, Calle 57 Nº 2216 - **TRENQUE LAUQUEN:** COMPUQUEN, Villegas 231 - **CORDOBA:** AUTODATA, Pasaje Santa Catalina 27 - TECSIEM, Santa Rosa 715 - **ROSARIO:** 2001 COMPUTACION, Santa Fe 1468 - MINICOMP, Maipú 862 - SISOR, Urquiza 1062 - **SANTA FE:** ARGECINT, P. San Martín 2433, L. 36 - SISOR, Rivadavia 1062 - INFORMATICA, San Gerónimo 2721/25 - **VILLA MARIA:** JUAN CARLOS TRENTO, 9 de Julio 80 - **MENDOZA:** INTERFACE, Sarmiento 98 - BIT & BYTE, 9 de Julio 1030 - **COMODORO RIVADAVIA:** COMPUSER, 25 de Mayo 827 - **GENERAL ROCA:** DISTRIBUIDORA VECCHI, 25 de Mayo 762 - **LA PAMPA:** MARINELLI, Pellegrini 155 - **NEUQUEN:** MEGA, Perito Moreno 383 - EDISA, Roca esq. Fotheringham - **RIO GRANDE:** INFORMATICA M & B, Perito Moreno 290 - **SAN CARLOS DE BARILOCHE:** L. ROBLEDO & ASOCIADOS, Elfein 13, Piso 1º - **TRELEW:** SISTENOVA, Sarmiento 456 - **PARANA:** MARIO GARCIA, Laprida y Santa Fe - **POSADAS:** CENTRO DE COMPUTOS ELDORADO, Colón 2429 - **RESISTENCIA:** FRANCO SANTI, Carlos Pellegrini 761 - **SAN SALVADOR DE JUJUY:** DELTA COMPUTACION, Salta 1108 - **SALTA:** DELTA COMPUTACION, Caseros 873 - **SAN MIGUEL DE TUCUMAN:** LEXICON, 9 de Julio 85 - ELECTRONICA